Cincom

AD/ADVANTAGE

MANTIS Application Development Tutorial
OpenVMS/UNIX

P39-1340-00

AD/Advantage® MANTIS Application Development Tutorial OpenVMS/UNIX

Publication Number P39-1340-00

© 2001 Cincom Systems, Inc.
All rights reserved

This document contains unpublished, confidential, and proprietary information of Cincom. No disclosure or use of any portion of the contents of these materials may be made without the express written consent of Cincom.

The following are trademarks, registered trademarks, or service marks of Cincom Systems, Inc.:

AD/Advantage®
C+A-RE™
CINCOM®
Cincom Encompass®
Cincom Smalltalk™
Cincom SupportWeb®
CINCOM SYSTEMS®
gOOi™
*i*D CinDoc™
*i*D CinDoc Web™
*i*D Consulting™
*i*D Correspondence™
*i*D Correspondence Express™
*i*D Environment™
*i*D Solutions™
intelligent Document Solutions™
MANTIS®
Mindspeed™
MindspeedXML™
SPECTRA™
SUPRA®
SUPRA® Server
Visual Smalltalk®
VisualWorks®

All other trademarks are trademarks or registered trademarks of:

Acucobol, Inc.
AT&T
Compaq Computer Corporation
Data General Corporation
Gupta Technologies, Inc.
International Business Machines Corporation
JSB Computer Systems Ltd.
Micro Focus, Inc.
Microsoft Corporation
Systems Center, Inc.
TechGnosis International, Inc.
The Open Group
UNIX System Laboratories, Inc.

or of their respective companies.

Cincom Systems, Inc.
55 Merchant Street
Cincinnati, Ohio 45246-3732
U. S. A.

PHONE: (513) 612-2300
FAX: (513) 612-2000
WORLD WIDE WEB: http://www.cincom.com

Attention:

Some Cincom products, programs, or services referred to in this publication may not be available in all countries in which Cincom does business. Additionally, some Cincom products, programs, or services may not be available for all operating systems or all product releases. Contact your Cincom representative to be certain the items are available to you.

Release information for this manual

The *AD/Advantage MANTIS Application Development Tutorial OpenVMS/UNIX*, P39-1340-00, is dated February 12, 2001. This document supports Release 2.8 of MANTIS.

We welcome your comments

We encourage critiques concerning the technical content and organization of this manual. Please take the survey provided with the online documentation at your convenience.

Cincom Technical Support for AD/Advantage

FAX: (513) 612-2000
Attn: AD/Advantage Support

E-mail: helpna@cincom.com

Phone: 1-800-727-3525

Mail: Cincom Systems, Inc.
Attn: AD/Advantage Support
55 Merchant Street
Cincinnati, OH 45246-3732
U. S. A.

Contents

About this book

Using this document

This tutorial provides you with a basic introduction to designing MANTIS® screens, files, and prompters, and to creating programs using the MANTIS language. For more detailed information on these topics, refer to AD/Advantage MANTIS Facilities OpenVMS/UNIX, P39-1300 and AD/Advantage MANTIS Language OpenVMS/UNIX, P39-1310.

This manual is not intended to teach you how to program, but rather to show you the basics of MANTIS so you can apply them to your data processing environment. Exercises appear at the end of each chapter. Take your time with the lessons and follow directions carefully.

This manual will guide you through most of the steps necessary to create an application in MANTIS. At each step, you will use MANTIS to create part of the application, and the exercises will teach you about MANTIS and its capabilities. Upon completion, you should be ready to begin writing or maintaining your own MANTIS applications.

Many of the MANTIS programming statements and commands appear in this tutorial, along with their syntax. “Conventions” on page xii describes the conventions that are used in the syntax notation. Read this section carefully before you begin the programming lessons in this tutorial. For more detailed information on programming statements and commands, refer to AD/Advantage MANTIS Language OpenVMS/UNIX, P39-1310.

Document organization

The information in this manual is organized as follows:

Chapter 1—Introduction
Provides a brief description of MANTIS, fundamental rules for its use (including sign-on and sign-off instructions), and an introduction to the Burrys application scenario that is used as the basis for this tutorial.

Chapter 2—Creating a screen
Describes the screen design process and shows you how to design the first screen in the Burrys scenario. The other three screens required for the Burrys application are presented as exercises.

Chapter 3—Creating a MANTIS file
Describes the file design process and shows you how to design the first file of the Burrys scenario. The other file required for the Burrys application is presented as an exercise.

Chapter 4—Creating a prompter
Describes the prompter design process and shows you how to design the prompter required for the Burrys scenario.

Chapter 5—Understanding MANTIS programming fundamentals
Explains MANTIS language conventions and shows you how to enter a program in an external editor. The exercises show you how to run a program to insert records into a file.

Chapter 6—Using MANTIS programming statements and commands
Describes the MANTIS programming environment in more detail, explains basic MANTIS programming statements, and shows you how to write the menu program for the Burrys application. In the exercises, you create the stubs for the browse and customer entry programs.

Chapter 7—Creating a browse program
Describes how to complete a program that displays the contents of the Burrys customer file. In the exercises, you write a program to browse the state codes file.

Chapter 8—Creating a data entry program
Shows you how to complete a program that validates and inserts new customer records into the customer file. In the exercise, you design a maintenance program that reads particular records from the Burrys customer file and deletes or updates them.

Chapter 9—What's next?
Summarizes the tutorial and provides references to further information on topics of interest.

Appendix A—Exercise examples
Provides examples of solutions for the exercises in this tutorial.

Index

Revisions to this manual

This new manual was added to the MANTIS documentation set for MANTIS Special Edition.

Conventions

The following table describes the conventions used in this document series:

Convention	Description	Example

`Constant width type`	Represents screen images and segments of code.	`Screen Design Facility`
		`GET NAME LAST`
		`INSERT ADDRESS`

Slashed b (ƀ)	Indicates a space (blank). The example indicates that a password can have a trailing blank.	`WRITEPASSƀ`
Brackets [ ]	Indicate optional selection of parameters. (Do not attempt to enter brackets or to stack parameters.) Brackets indicate one of the following situations.	
	A single item enclosed by brackets indicates that the item is optional and can be omitted. The example indicates that you can optionally enter a program name.	`COMPOSE [`*`program-name`*`]`
	Stacked items enclosed by brackets represent optional alternatives, one of which can be selected. The example indicates that you can optionally enter NEXT, PRIOR, FIRST, or LAST. (NEXT is underlined to indicate that it is the default.)	[**NEXT** **PRIOR** **FIRST** **LAST**]

Convention	Description	Example
Braces { }	Indicate selection of parameters. (Do not attempt to enter braces or to stack parameters.) Braces surrounding stacked items represent alternatives, one of which you must select. The example indicates that you must enter FIRST, LAST, or a value for *begin*.	**FIRST** ***begin*** **LAST**
Underlining (In syntax)	Indicates the default value supplied when you omit a parameter. The example indicates that if you do not specify ON, OFF, or a row and column destination, the system defaults to ON.	**SCROLL** [**ON** **OFF** [***row***][,***col***]]
	Underlining also indicates an allowable abbreviation or the shortest truncation allowed. The example indicates that you can enter either PRO or PROTECTED.	PROTECTED
Ellipsis points...	Indicate that the preceding item can be repeated. The example indicates that you can enter (A), (A,B), (A,B,C), or some other argument in the same pattern.	(*argument*,...)

Convention	Description	Example
UPPERCASE	Indicates MANTIS reserved words. You must enter them exactly as they appear. The example indicates that you must enter CONVERSE exactly as it appears.	`CONVERSE name`
Italics	Indicate variables you replace with a value, a column name, a file name, and so on. The example indicates that you can supply a name for the program.	`COMPOSE [program-name]`
Punctuation marks	Indicate required syntax that you must code exactly as presented. ( ) parentheses . period , comma : colon ; semicolon ' single quotation mark " " double quotation marks	[LET]$_V$ $\begin{bmatrix}(i)\\(i,j)\end{bmatrix}$ [ROUNDED(n)] = $e1$ [, $e2, e3 \ldots$]

MANTIS documentation series

MANTIS is an application development system designed to increase productivity in all areas of application development, from initial design through production and maintenance. MANTIS is part of AD/Advantage®, which offers additional tools for application development.

The AD/Advantage/MANTIS manual series provides documentation for system planning, system administration, object design, object generation, and programming. The series also includes reference manuals for language statements and tutorial manuals on object design and programming. The manuals available are listed below with a brief description.

Document	Publication No.	Description
AD/Advantage MANTIS 2.8.x Installation and Startup OpenVMS/UNIX	P39-0027	Describes the installation and startup procedures for a the latest and release of MANTIS. Also includes information for some previous releases and patches.
AD/Advantage MANTIS Facilities OpenVMS/UNIX	P39-1300	Describes using MANTIS design facilities in the PC environment, including transferring entities user-to-user and system-to-system, and prototyping. Also provides information on migrating OpenVMS/UNIX applications to and from the mainframe side.
AD/Advantage MANTIS Language OpenVMS/UNIX	P39-1310	Discusses MANTIS programming fundamentals in the OpenVMS/UNIX environment and lists MANTIS commands, statements, and functions.
AD/Advantage MANTIS Administration OpenVMS/UNIX	P39-1320	Guides the Master User in customizing and maintaining the OpenVMS/UNIX development environment.

Document	Publication #	Description
AD/Advantage MANTIS Messages and Codes OpenVMS/UNIX	P39-1330	This manual provides explanations and actions for all messages you may encounter while using MANTIS and AD/Advantage.
AD.Advantage MANTIS Application Development OpenVMS/UNIX	P39-1340	This manual provides a basic introduction to designing MANTIS screens, files, and prompters, and to creating programs using the MANTIS language.
AD/Advantage MANTIS Supra SQL Programming OpenVMS/UNIX	P39-1345	Discusses embedding SUPRA SQL in MANTIS statements, and binding SQL programs. Guides the Master User in customizing and maintaining the MANTIS SUPRA SQL environment.
AD/Advantage MANTIS Rdb Programming Open VMS	P39-1350	Discusses embedding SUPRA SQL in MANTIS statements, and binding SQL programs. Guides the Master User in customizing and maintaining the MANTIS Rdb environment.
AD/Advantage MANTIS Oracle Programming UNIX	P39-1355	Discusses embedding SQL in MANTIS statements in the MANTIS Oracle environment. Guides the Master User in customizing and maintaining the MANTIS Oracle environment.

Educational material

AD/Advantage and MANTIS educational material is available from your regional Cincom education department.

1

Introduction

MANTIS is a comprehensive application development system designed to increase productivity in all areas of application development—from initial design through production.

MANTIS offers design facilities, prototyping capabilities, testing and debugging tools, and an advanced, high-level programming language. Among other things, MANTIS enables you to:

- Design and create screens that display data attractively by taking full advantage of large screen capabilities and the features available on today's terminals (such as color, reverse video, blinking, etc.)
- Design and create permanent files for data storage and manipulation
- Create and test programs interactively using structured programming concepts

All MANTIS facilities are completely interactive. This means that once a program, screen, or file is created, it is immediately available for display and review by end users. This eliminates the need for precompiling, compiling, binding, coding Job Control Language, and other activities normally associated with application development.

This tutorial will show you how to use the MANTIS screen, file, and prompter design facilities. It will also demonstrate how you can use the MANTIS programming language and the Program Design Facility to quickly and easily build programs that meet your application development needs.

Before you begin

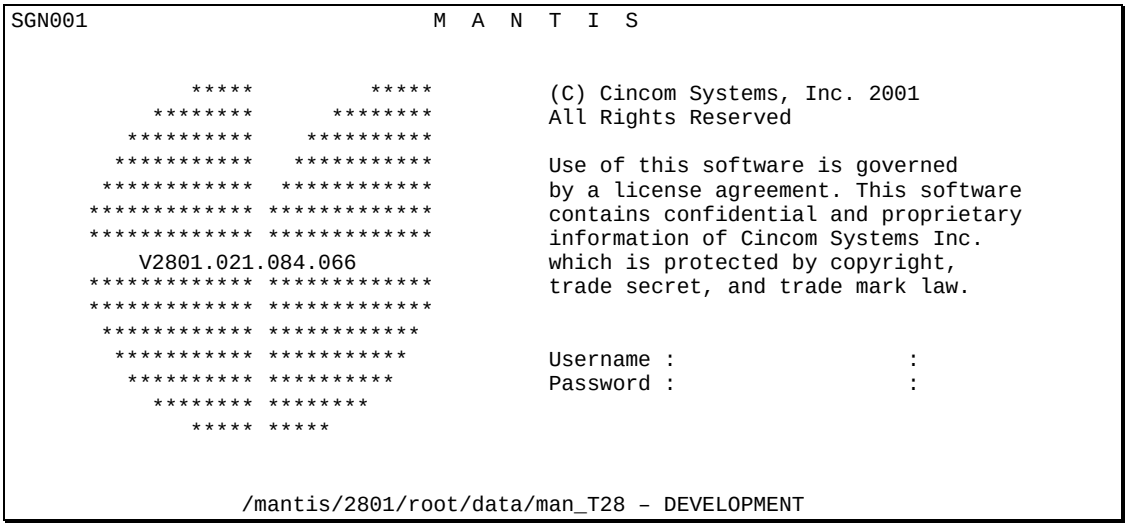
Each MANTIS installation has a person (or persons) designated as the MANTIS Master User. The Master User has access to certain facilities and information not available to all MANTIS users.

You will normally access MANTIS from a menu program that already exists at your site. Check with your Master User for instructions. Your Master User will also supply you with a valid user ID and password to sign on to MANTIS.

Signing on to MANTIS

To sign on to MANTIS, obtain the valid transaction ID for MANTIS, a valid user ID, and a valid password from your Master User.

When you enter the transaction ID, the sign-on screen appears as shown in the following screen (unless your Master User has changed the Cincom logo to another display):



Enter your valid user ID and password; then, press ENTER. You are signed on to MANTIS and your Facility Selection menu appears.

The following screen illustration shows the standard Facility Selection menu that is provided with MANTIS:

```
FAC002                          M  A  N  T  I  S


                               FACILITY  SELECTION



   Run A Program  ..............  1        Sign On As Another User  ....  11
   Display A Prompter  .........  2        MANTIS Run System  ..........  12
   Design A Program  ...........  3        Run A Scenario  .............  13
   Design A Screen  ............  4        Directory Facility  .........  14
   Design A File Profile  ......  5        Transfer Facility  ..........  15
   Design A Prompter  ..........  6        Universal Export Facility  ..  16
   Design An Interface  ........  7
   Design An Ultra File View  ..  8
   Design An External File View   9        Spectra  ....................  19
   Design A Scenario  .......... 10        Search Facility  ............  20
                                           Exit MANTIS  ............  CANCEL

                                    :    :
```

Your user ID name displays directly below the MANTIS Facility Selection Menu heading. (The illustrations throughout this tutorial show BURRYS as the user ID.)

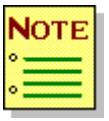
The Facility Selection menu lists the facilities that are available for your MANTIS installation. Your Master User may have omitted some of these facilities, and/or added new facilities to meet your specific needs.

The cursor appears in the action field. To access a facility from the menu, enter the number of the facility in the action field, and press ENTER.

Signing off MANTIS

To sign off, press CANCEL until you reach the Facility Selection menu. Your Master User will provide you with directions for signing off MANTIS from this point.

On a VT terminal, the PF1 and – keys are equivalent to the CANCEL key, which returns you to a higher screen. However, the CANCEL key may be different for your installation. Check with your Master User if you are not sure which key to use as the CANCEL key.



You can sign off at any time during the tutorial session. But, before you do, be sure to save or replace all design work through the library functions in the facility where you are currently working.

If you forget to save your work, you will receive a message that unsaved changes exist.

When you are ready to begin again, use the start-up procedure in “Signing on to MANTIS” on page 18. Then, if you are continuing work on an existing design, use the library functions of the corresponding design facility to fetch the design into your work area.

Introduction to the Burrys scenario

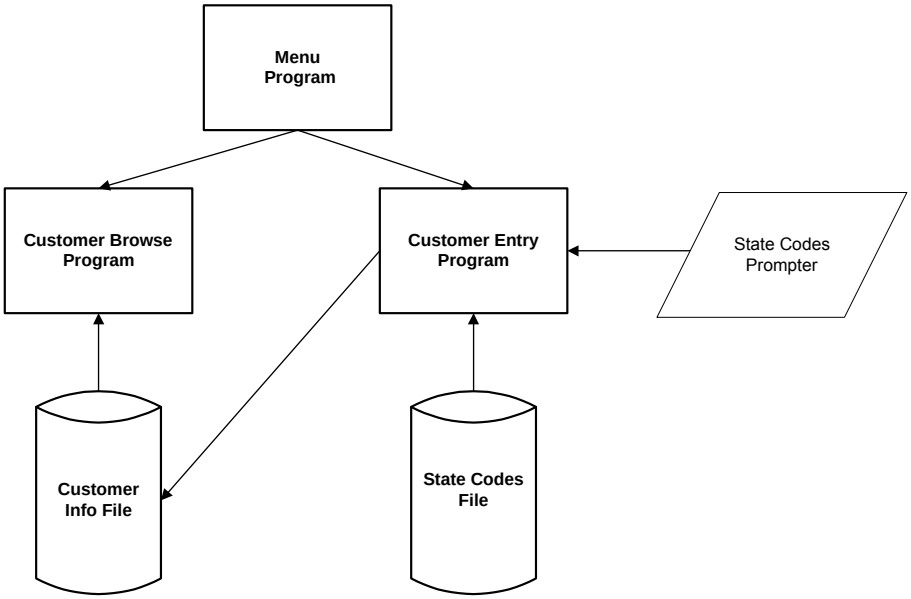
In this tutorial, you will use the MANTIS facilities to create an application for a fictitious corporation named “Burrys,” an electrical goods retailer with branches in various parts of the country.

Managers at each branch are currently unable to establish credit for new customers without first contacting the corporate accounting department. The accounting department establishes a customer’s credit rating and credit limit, and assigns a customer number.

Over the past two years, however, Burrys has increased its sales by 135% and its customer base by 53%. Branch managers all agree that the present system for customer file maintenance and inquiry is inadequate.

The branch managers have requested, and received approval for, an online, automated system that will allow them to access the corporate database. The system will enable managers to request customer information and add new customers to the database.

The following diagram shows the Burrys customer accounts system:



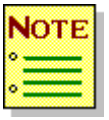
Burrys has chosen MANTIS as its application development system. The prototyping capability and ease of use of MANTIS will minimize the time and cost of developing and implementing the system.

In the following chapters, you will develop the Burrys customer accounts system. The system will require:

- Four screens:
  - Menu (CUST_MENU)
  - Customer entry (CUST_ENTRY)
  - Customer browse (CUST_BROWSE)
  - State code entry (STATE_CODE)

- Two files:
  - Customer information (CUST_INFO)
  - State codes (STATE_CODES)
- A state codes prompter (STATE_CODES)
- Three programs:
  - **Menu (CUST_MENU).** Displays the menu screen and allows the user to choose the customer entry and browse programs, or exit from the Burrys system
  - **Customer entry (CUST_ENTRY).** Validates the information provided about a customer and inserts the data into the customer information file
  - **Customer browse (CUST_BROWSE).** Displays the contents of the customer information file on the customer browse screen

In the Design Facilities portion of this tutorial (chapters 2 through 4), you will create the screens, files, and prompter. Once you have completed these items, you will use the Program Design Facility and an external editor to create and run the menu, customer entry, and customer browse programs.



The examples in this tutorial show MANTIS keys that are standard for a VT terminal. Certain key assignments may differ for your environment, depending on how your Master User has configured MANTIS, and upon the terminal emulator you are using and how it has been configured.

2

Creating a screen

Before you begin creating the Burrys application, you'll need to know some basic features of the MANTIS Screen Design Facility. This chapter introduces you to these concepts, then guides you step-by-step through the process of creating the customer browse screen for the Burrys customer accounts application. The other three screens required for the application are presented as exercises.

Learning outline

In this chapter you will learn how to:

- Use the screen design work area
- Access the Screen Design Facility menu and select options on it
- Design a screen by creating and formatting headings and data fields
- Save a new screen, or changes to an existing screen
- Specify the data type and other attributes for a screen field
- View the attributes that you have specified for screen fields
- Assign repeat specifications to screen fields
- Display a completed screen design
- View the directory of screens
- Print the completed screen design
- Use window mode to create and view screens that are larger than the physical display

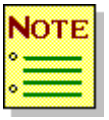
Basic concepts: Understanding screen design

Before you begin designing your screens for the Burrys application, let's take a look at some basic concepts used in the MANTIS screen design process.

Using the screen design work area

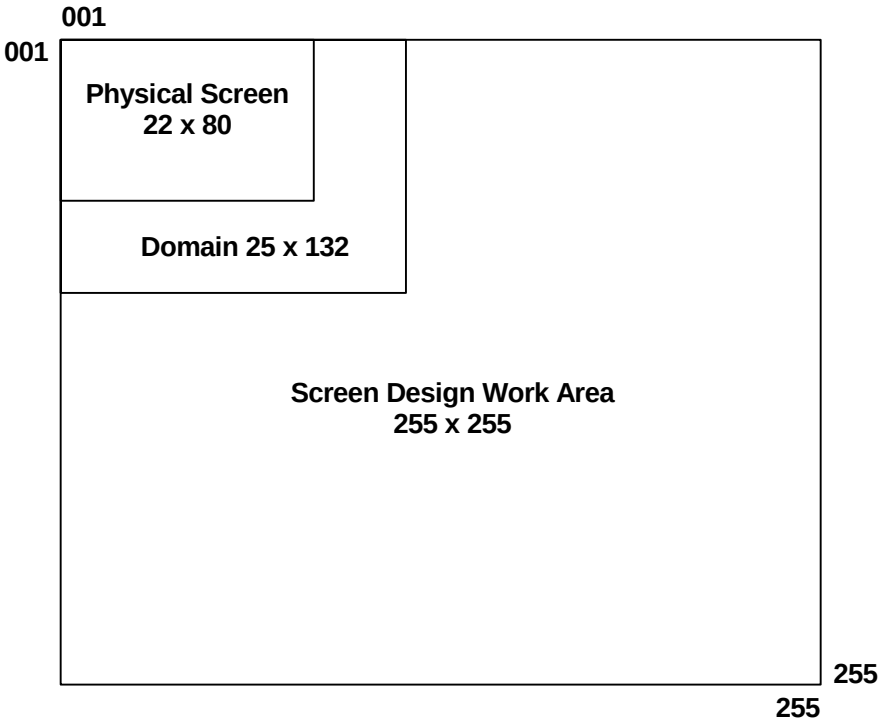
MANTIS provides a screen design work area that can be up to 255 rows by 255 columns. The size of this work area determines the size or domain of your screen. You may adjust the size of your screen at any time during the Create or Update process.

Since your screen design can be larger than your physical screen, use your physical screen as a movable window to view different sections of the display.



Screen designs start in row 1 column 1, but when you use them in a program, you may place them anywhere in the logical display (32767 rows by 32767 columns) using CONVERSE statement. The CONVERSE statement is documented in AD/Advantage MANTIS Language OpenVMS/UNIX, P39-1310.

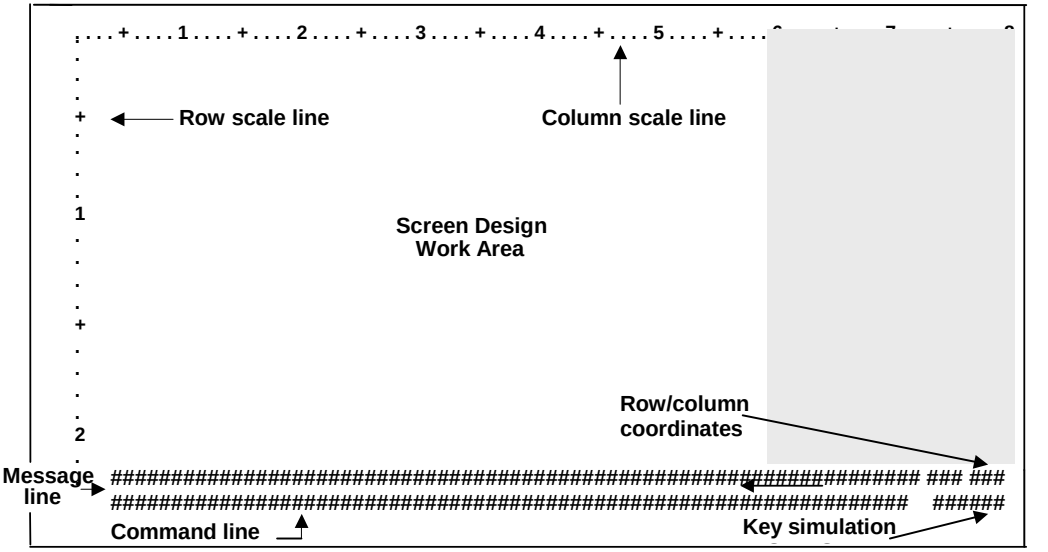
Each screen that you create has a domain within the work area. For example, in this illustration the screen domain is 25 rows long and 132 columns wide:



The first set of Burrys screens that you design will use a 22x80 screen domain, so the logical and physical screen sizes are the same for a 24x80 terminal. MANTIS reserves the bottom two lines for its use (see below).

If the row scale line is turned off during screen design, MANTIS reserves the first column of the screen. If the row scale line is turned on, MANTIS reserves the first column of the screen. The column that MANTIS reserves is not part of the screen design work area.

The last two physical lines of the screen design work area are reserved for the message, row/column coordinates (not displayed during the screen design painting session), command line, and key simulation field. These fields remain on your screen even when you move your physical screen around the work area:



The following table describes the reserved fields on the screen design work area:

Field	Length	Description
Message line	Terminal width minus 12 columns	Displays system messages.
Row/column coordinates	5 bytes for each	Displays the current row and column coordinates of the upper-left corner of the screen design (when in window mode).
Command line	Terminal width less 8 columns	Provides a field for entering unsolicited input. Your MANTIS program picks up data from this field with the OBTAIN statement.
Key simulation field	7 bytes	Provides a field for entering function keys, if your keyboard does not have them.
Row scale line	1 byte per line	Row scale ruler (when present). Always occurs on the left edge of the physical screen.
Column scale line	Terminal width	Column scale ruler (when present). May occur on any line in the design area.

The row and column scale lines (when displayed) indicate the current row and column position within the work area. These scale lines change to reflect your position as you move around the work area. They will not appear as part of your finished design, but are provided to assist you with your design layout.

Use PF3 to display, move, and remove the column scale line. Use PF9 to display and remove the row scale line.

In the Screen Design Facility, row and column coordinates appear automatically in the lower right corner of the screen only when you display a completed screen design that is outside the boundaries of the terminal (that is, the screen domain (logical terminal) is larger than the physical screen).

Using screen design commands

You can use your terminal screen as a moveable window, scrolling around the screen design to view different sections. To do so, you use MANTIS screen design commands to move the window and edit your screen design.

You specify these commands by using PF keys, or by entering a command on the command line. (Commands that you enter on the command line are described on page 31.) The following table lists the available PF keys, and describes how they work:

PF key	Function	Description
PF1	Insert a line	If you place the cursor before a field, MANTIS inserts a blank line at the current cursor position, provided that there is at least one empty line available in the specified screen domain. (Compare with PF7.)
PF2	Delete a line	If the cursor is in position 1 of the line, MANTIS deletes that line. Otherwise, MANTIS deletes all fields that start within 80 columns to the right of the cursor, including those fields outside the window boundary up to, but not including, the corresponding cursor position on the next line. Fields on following lines that are not deleted move up one line.
PF3	Insert, move, or delete the column scale line	Place the cursor on the line in the screen design work area where you want the column scale line to appear; then, press PF3 . The column scale line displays, temporarily overlaying any data that was displayed on this line. To clear the column scale line from the screen, place the cursor on the column scale line and press PF3 again. The data that was under the scale line reappears. You cannot place the column scale line on the last two lines of the physical screen.
PF4	Move a field	Position the cursor anywhere in the field you want to move and press PF4. Move the cursor to where the first character of the field should appear and press PF4 again. The field, along with its attribute information, moves to the new position. If you try to move a field that is too long to fit within the 255-column limit, MANTIS moves the field, but truncates the overflow.

PF key	Function	Description
PF5	Copy a field	Position the cursor anywhere in the field that you want to copy and press PF5. Then, move the cursor to the position where the first character of the copied field should appear, and press PF5 again. The field, along with its attribute information, is copied. If you try to copy a field that is longer than 255 columns, MANTIS copies the field, but truncates the overflow.
PF6	Delete a field	Position the cursor anywhere in the field that you want to delete and press PF6. MANTIS deletes only the field where the cursor is placed; other fields on the same line remain in the same location.
PF7	Insert a range of lines	Position the cursor anywhere on the line after which blank lines are to be inserted and press PF7. MANTIS prompts you for the number of lines to insert. Specify the appropriate number and press RETURN. If there are insufficient blank lines in the screen domain, MANTIS inserts as many lines as possible, then terminates the request.
PF8	Delete a range of lines	Position the cursor anywhere on the line where deletion is to start and press PF8. MANTIS prompts you for the number of lines to delete. Specify the required number and press RETURN. Note that unlike PF2, only complete lines are deleted; therefore, the position of the cursor within the selected line is not relevant.
PF9	Add or remove the row scale line	If the row Scale line is already displayed, it is removed; if not, it is displayed in column 1, temporarily overlaying any data in column 1.

PF key	Function	Description
PF10	Move a range of lines	Position the cursor anywhere on the first line to be moved and press PF10. MANTIS prompts you to select the last line in the range. Position the cursor anywhere on the last line and press RETURN. When MANTIS prompts you to select the destination line, position the cursor anywhere on the first line of the destination and press RETURN. Pressing any key other than RETURN at any point terminates the operation. MANTIS only moves the selected lines if the entire destination area is blank. The first and last lines selected may be the same line and the last line may precede the first line. The field names and attributes of all fields within the selected range are also moved during this process.
PF11	Copy a range of lines	This key works the same as PF10, except that the lines are copied instead of moved. The process produces fields with the same name. You must change one of each pair using the Update Field Specifications option.
PF12	Display the screen domain	Display the screen domain on the last line of the screen. You may change this value if you wish and press RETURN. If existing fields fall outside the new domain, MANTIS automatically extends the domain to the minimum size required to contain all fields.
PF13	Convert simulated box characters to box graphics.	Simulated box characters are only recognized when there are two or more of them in adjacent positions that could form part of a box. Simulation characters are defined by the **BOXCHARS** MANTIS option.
PF14	Convert box graphics to simulated box characters	See PF13 above for more information

PF key	Function	Description
PF15	Equivalent to GOLD/E	Invoke external editor
GOLD/H	Display a help screen	The Help screen summarizes the Screen Design PF keys and shows other information such as the current screen's domain size. The current screen's blank-fill and mask character may be updated via this screen.
GOLD/E	Invoke the external editor	Use the external editor to edit your screen design. You may want to use this editor when you create a new design.
MINUS	Draw boxes	Draw boxes using the cursor keys
COMMA	Erase boxes	Erase boxes using the cursor keys

Understanding screen fields

You can specify two types of fields for a screen design:

- **Data fields.** These fields are designated by a *mask character* (usually hash signs (#), unless your Master User has configured MANTIS to use another mask character). Data appears in these fields as it is exchanged with a running program. When you design a screen, you assign symbolic names to the data fields and specify attributes for the fields (for example, bright intensity).
- **Heading fields.** These fields may contain any valid MANTIS character except the mask character. These fields always appear on the final display exactly as you enter them during screen design, with the exception of the blank fill character (in our example, the vertical bar).

Step-by-step: Creating the customer browse screen

Now that you understand some basic concepts of screen design, you're ready to design your first MANTIS screen.

Step 1: Selecting the Screen Design Facility

When you sign on to MANTIS, the MANTIS Facility Selection menu appears:

```
FAC002                          M  A  N  T  I  S


                               FACILITY  SELECTION



   Run A Program  ..............  1        Sign On As Another User  ....  11
   Display A Prompter  .........  2        MANTIS Run System  ..........  12
   Design A Program  ...........  3        Run A Scenario  .............  13
   Design A Screen  ............  4        Directory Facility  .........  14
   Design A File Profile  ......  5        Transfer Facility  ..........  15
   Design A Prompter  ..........  6        Universal Export Facility  ..  16
   Design An Interface  ........  7
   Design An Ultra File View  ..  8
   Design An External File View   9        Spectra  ....................  19
   Design A Scenario  .......... 10        Search Facility  ............  20
                                           Exit MANTIS  ............  CANCEL

                                    :    :
```

The cursor is positioned in the action field, where you indicate your facility selection.

You want to design a screen, so select the Design a Screen facility by entering 4 in the action field and pressing ENTER. The Screen Design Facility menu appears, and the cursor is positioned in the load screen field :

```
SCR001                          M  A  N  T  I  S

                            SCREEN DESIGN FACILITY

                     Create or Update a Screen  ............  1
                     Update Field Specifications  ..........  2
                     List Field Specifications  ............  3
                     Update Repeat Specifications  .........  4
                     List Repeat Specifications  ...........  5
                     Display Completed Design  .............  6
                     Library Functions  ....................  7
                     Directory Of Screens  .................  8
                     Print Completed Design  ...............  9
                     Exit Screen Design  ..............  CANCEL

   Load Screen .. :                                                    :
   Current Screen :

                                       :    :
```

Because you are designing a new screen, you will follow the sequence of functions as they are listed on the Screen Design Facility menu (that is, first create a screen, then update field specifications, etc.).

You can select a design facility menu option in one of two ways: enter the corresponding number in the action field and press ENTER, or press the corresponding PF key.

Step 2: Creating and formatting screen fields

In this step, you will create and format all first occurrences of your screen fields. To begin, select the Create or update a screen option by entering 1 in the action field and pressing ENTER, or by pressing PF1. Then, press ENTER to accept the default screen domain size of 22 rows by 80 columns. Without moving the cursor, press PF3 to display the column scale line, as shown in the following screen:

```
 -...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
-
-
-
+
-
-
-
-
1
-
-
-
-
+
-
-
-
-
2
-
-
 .
```

Remember that your screen design may be as large as 255 rows by 255 columns. You can scroll around the work area by pressing application keypad 8 (up), 2 (down), 4 (left), or 6 (right). Refer to AD/Advantage MANTIS Administration OpenVMS/UNIX, P39-1320, for more detailed information about keyboard interpretation, and how you can effectively customize your keyboard for MANTIS.

The cursor is positioned in the second column of the screen because MANTIS reserves the first column . If you try to enter data in these columns, MANTIS will respond with an alarm bell. You can press PF9 to remove the row scale line.

For this screen, you will define both heading and data fields. Remember, heading fields appear in the final display exactly as you enter them in screen design, except for the blank fill character (in this case, the vertical bar).

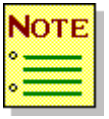
Place the cursor anywhere on line three and press PF3 to reposition the column scale line. Enter the heading fields as shown below, using a vertical bar to separate letters and words as indicated:

```
.                                B|U|R|R|Y|S
.                               CUSTOMER|BROWSE
. ....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+...
   CUSTOMER|||||||CUSTOMER|||||||BRANCH|||||||CREDIT|||||||CREDIT
+   NUMBER|||||||||||NAME|||||||||NUMBER|||||||RATING|||||||||LIMIT
.
.
.
.
1
.
.
.
.
+
.
.
.
.
2
.
.
```

If you make an error while entering your design, back space and type over the mistake with correct data, or erase the incorrect data with the space bar.

Because MANTIS interprets a blank (␢) as designating a new field, you should enter a blank fill character (in this case, the vertical bar character) between the words and letters of heading fields. The blank fill character connects words and letters into one field so that data is transmitted more efficiently. (For example, the heading B␢U␢R␢R␢Y␢S assumes five heading fields, but B|U|R|R|Y|S assumes only one heading field.) When the screen is displayed, the vertical bars do not appear. (The blank fill character is changeable on the library functions screen, so that you can display the vertical bar if you wish.)

When you finish entering the heading fields, press ENTER. MANTIS checks your entry for errors and temporarily stores the design in your work area.



MANTIS may display an error message if an error is made at any point in the screen design process. For more information on messages, refer to AD/Advantage MANTIS Messages and Codes OpenVMS/UNIX, P39-1330.

Using the information in the table below the screen, add two lines of data fields (designated by hash signs):

```
                                  B|U|R|R|Y|S
                                CUSTOMER|BROWSE
 ....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+...
   CUSTOMER|||||||CUSTOMER||||||||BRANCH|||||||CREDIT||||||||CREDIT
+   NUMBER|||||||||||NAME|||||||||NUMBER|||||||RATING|||||||||LIMIT

    ######  ####################  ####           ##          $#####
1
2
 ################################################################################
```

Field	Field length
CUSTOMER NUMBER	6
CUSTOMER NAME	20
BRANCH NUMBER	4
CREDIT RATING	2
CREDIT LIMIT	6 (includes the edit character “$”)
MESSAGE	76

Notice that instead of using vertical bars between fields as you did with heading fields, you now enter blank spaces after each data field.

Note two other points about the preceding screen:

- You are entering an edit character ($) under CREDIT LIMIT. You may use any valid MANTIS character as an edit character except the mask character (#, or some other character designated on the library functions screen). You may also use any character that you can enter from the keyboard (for example, national characters, punctuation, etc.)

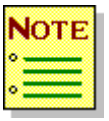
  Edit characters display only when there is data in the field. (Edit characters are used by numeric data types only. Information on text vs. numeric data fields will be explained in chapter 5 of this tutorial.)

- You are designating a data field for messages across the bottom of the screen. When the screen is used in a program, this field can display help messages, policy reminders, or any other information a user might need to interpret the screen.

When you have finished entering the data fields, press ENTER.

You have now defined all the heading fields and the first occurrences of the data fields for this screen. Press CANCEL to return to the Screen Design Facility menu:

```
SCR001                          M  A  N  T  I  S


                              Screen Design Facility


                     Create or update a screen  ............  1
                     Update field specifications  ..........  2
                     List field specifications  ............  3
                     Update repeat specifications  .........  4
                     List repeat specifications  ...........  5
                     Display completed design  .............  6
                     Library functions  ....................  7
                     Directory of screens  .................  8
                     Print completed design  ...............  9
                     Exit Screen Design  ..............  CANCEL

   Load Screen .. :                                                    :
   Current Screen :


                                       :   :
```



You can move among the items listed on the Screen Design menu without losing your screen design, as long as you press ENTER before exiting from each option. You *must* use the screen design library functions to save your screen design before you exit to the MANTIS Facility Selection menu, or you will lose your work.

Step 3: Using the screen design library functions

To exit from MANTIS at this point, you must first save your screen design. (If you want to proceed with the tutorial lesson, you do not have to save your design now.)

To save your design, enter 7 in the action field at the bottom of the Screen Design Facility menu and press ENTER. The Screen Design Library Facility screen displays:

```
SCR002                          M  A  N  T  I  S

                        SCREEN DESIGN LIBRARY FACILITY

     Screen Name ....... :                                                  :
     Description ....... :                                                 :
     Language .......... : ENGLISH         :
     Screen Size ....... : 24 x 80  : (Rows x Columns)
     Export File Name .. :                                  :

     Save  ...................  1           Sound Alarm ........... : N :
     Replace  ................  2           Full Display .......... : N :
     Fetch  ..................  3           Protect Bottom Line ... : N :
     Delete  .................  4           Field Separators ...... : N :
     Clear Work Area  ........  5           Opaque Map ............ : N :
     Export  .................  6           Automatic Windowing ... : Y :
     Import  .................  7           Horizontal Windowing .. : Y :
                                            Vertical Windowing .... : Y :
     Exit  ..............  CANCEL           Mask Character ........ : # :
                                            Blank Fill Character .. : | :
                     :   :
```

The cursor is positioned at the beginning of the Name of screen field. Enter a symbolic name for the screen design: CUST_BROWSE.

A screen name can be up to 30 characters in length.

Now, move the cursor to the beginning of the Description field. Supply a description of your screen: BURRYS CUSTOMER BROWSE SCREEN. This description helps you to identify the screen in your directory.

Keep names and descriptions short, accurate, and consistent for easy reference and use.

Let the other fields assume their default values:

Field name	Description	Default value
Screen Size	Row and column specifications for the screen based on the size of the physical terminal (or terminal emulator) where you plan to display the screen. The following screen sizes are allowed: 24x80, 24x132, 43x80,32x80,27x132..	24 x 80
Language	Specifies the language attribute for the screen design. MANTIS uses this to find a screen that matches the running user's specified language.	English
Blank fill character	Specifies the character you want to use to fill blank spaces	Vertical bar (|)
Mask character	Specifies the character you choose to use as the mask character to identify fields	#
Sound alarm	Indicates that you want to sound an alarm (beep) each time you converse the screen	No
Full display	Expands the screen size to the dimensions of the current terminal, including the bottom two lines of the screen	No
Protect bottom line	Protects the bottom line of the screen (the line that contains the command line and key simulation fields)	No

Field name	Description	Default value
Opaque map	Indicates whether or not a screen (map) will be opaque (rather than transparent) when it is displayed	No
Automatic Windowing	Indicates whether you want MANTIS to allow entry for any field that is only partially displayed on the screen and to allow MANTIS to automatically window your screen when tabbing to a field that is not displayed on your physical terminal.	Yes
Horizontal Windowing	Indicates whether or not a screen (map) will move horizontally when a windowing operation is performed.	Yes
Vertical Windowing	Indicates whether or not a screen (map) will move vertically when a windowing operation is performed.	Yes
Field Separators	Indicates whether or not horizontally repeated field occurrences must be separated by a field separator (at least one blank).	Yes

Your screen should look like this one:

```
SCR002                          M  A  N  T  I  S

                        SCREEN DESIGN LIBRARY FACILITY

     Screen Name ....... : CUST_BROWSE                                      :
     Description ....... : Burrys customer browse screen                   :
     Language .......... : ENGLISH         :
     Screen Size ....... : 24 x 80  : (Rows x Columns)
     Export File Name .. :                                  :

     Save  ...................  1          Sound Alarm ........... : N :
     Replace  ................  2          Full Display .......... : N :
     Fetch  ..................  3          Protect Bottom Line ... : N :
     Delete  .................  4          Field Separators ...... : N :
     Clear Work Area  ........  5          Opaque Map ............ : N :
     Export  .................  6          Automatic Windowing ... : Y :
     Import  .................  7          Horizontal Windowing .. : Y :
                                           Vertical Windowing .... : Y :
     Exit  ..............  CANCEL          Mask Character ........ : # :
                                           Blank Fill Character .. : | :
               : 1  :
```

Select the Save option by entering 1 in the action field. Press ENTER. MANTIS automatically returns you to the Screen Design Facility menu and displays a confirmation message in the bottom, left corner of the screen:

```
SCR001                          M  A  N  T  I  S


                              Screen Design Facility


                     Create or update a screen  ............  1
                     Update field specifications  ..........  2
                     List field specifications  ............  3
                     Update repeat specifications  .........  4
                     List repeat specifications  ...........  5
                     Display completed design  .............  6
                     Library functions  ....................  7
                     Directory of screens  .................  8
                     Print completed design  ...............  9
                    Exit Screen Design  ..............  CANCEL

   Load Screen .. :                                                :
   Current Screen :

                                       :   :
  'CUST_BROWSE' SAVED
```

To take a break in the lesson at this point, press CANCEL to exit to the MANTIS Facility Selection menu; then, press CANCEL again to exit MANTIS.

If you exit the Screen Design Facility at this point, you will need to fetch your screen design when you are ready to continue with the screen design lesson. To do so:

1. Select the Design a Screen option from the MANTIS Facility Selection menu.

2. Select the Library functions option.

3. Enter the name of your screen design (CUST_BROWSE).

4. Select the Fetch option (enter 3 in the action field and press ENTER). MANTIS returns you to the Screen Design Facility menu and displays a confirmation message in the bottom, left corner of the screen. The screen is now in your work area, and ready for modification.

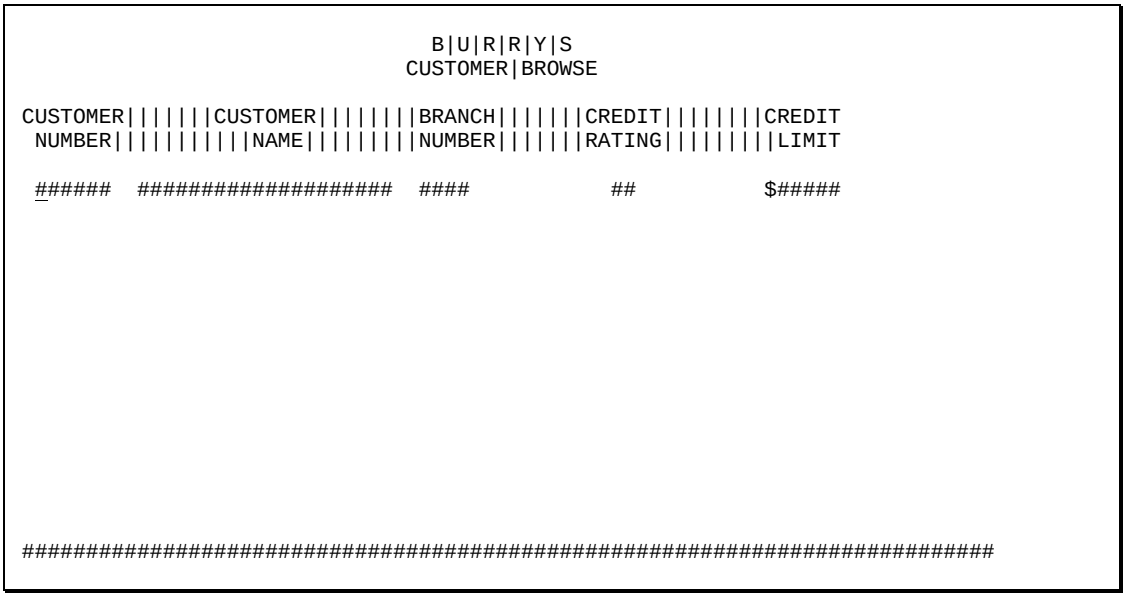
Step 4: Updating field specifications

You must now provide specifications (characteristics) for the data fields you created. The cursor is positioned in the action field on the Screen Design Facility menu. To select Update field specifications, enter 2 in the action field and press ENTER. MANTIS displays options of inspect or alter field attributes.

```
SCR003                          M  A  N  T  I  S


                        INSPECT OR ALTER FIELD ATTRIBUTES


                  Next Undefined Field  .......................  1
                  Nominate By Positioning Cursor On Field  ....  2
                  Use Cursor To Select A Range Of Fields  .....  3
                  Set Common Attributes For All Box Fields  ...  4
                  Set Common Attributes For All Headings  .....  5
                  Set Common Attributes For All Data Fields  ..  6
                  Supply Symbolic Name Of Field  ..............  XXX...
                  Exit  .......................................  CANCEL


                         :                                  :
```

To define or alter field specifications, you can select a field in one of three ways:

- Next Undefined Field
- Nominate by Positioning Cursor on Field and press ENTER.
- If you have already provided a name for the field, you can enter the field name on the command line and press ENTER.



- Because you are creating a new screen, all of your data fields are undefined (that is, they do not have symbolic names or attributes). Move the cursor to the customer number data field and press ENTER. When you select a field, MANTIS highlights the field and displays a window for you to define field specifications:

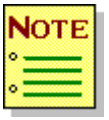
```
                              B|U|R|R|Y|S
                             CUSTOMER|BROWSE

 CUSTOMER|||||||CUSTOMER|||||||||BRANCH|||||||CREDIT|||||||||CREDIT
  NUMBER||||||||||||NAME|||||||||NUMBER|||||||RATING|||||||||LIMIT

  ######  ####################  ####           ##           $#####




+-------------------------------------------------------------------------------+
! Field Name    : cust_number                     :  Row/Column   : 7   : 3   : !
! Intensity     : NORMAL :     Data Type       : TEXT    :   Auto-Skip   : Y : !
! Reverse/FF    : N : N :      Protected       : N :         Blinking    : N : !
! Underline/FF  : N : N :      Prot Input Only : N :         Uppercase   : N : !
! Extended      : N :          Length          : 6   :       Cursor      : N : !
! Color         :           :  Detectable      : N :         Modified    : N : !
! Left Bar      : N :          Right Bar       : N :         Over Bar    : N : !
! Double Height : N :          Double Width    : N :         Highlight   : N : !
! Vert Rep/Disp : 14  : 1   :  Horiz Rep/Disp  :     :     : Message     : N : !
! Entry Routine :                                 :                         !
!                               Field Sensitive Validation / Forced  : N : N : !
+-------------------------------------------------------------------------------+
```

The customer number field is highlighted and a boxed prompt lists the current (or default) attribute and repeat specifications for the field. MANTIS also displays the row/column coordinates and the length of the field.

You need to define the remaining attributes for this field. Make the following changes to the attribute specifications for the customer number field as shown in the preceding screen illustration. To do so, place the cursor in the field and enter the indicated data:

- **Field Name (CUST_NUMBER).** Provides a name that identifies the data in this field. You can assign names to data fields only; not to heading fields. (For heading fields, the name is filled with a NOT REQUIRED message.). The symbolic name must follow the rules for MANTIS symbolic names (see “Using symbolic names” on page 108). The symbolic name must begin with an alphabetic character. It can contain 1-30 alphabetic and/or numeric characters, and the underline character *only*, and may not contain blank spaces or hyphens. (Note that the parts of the symbolic name CUST_NUMBER are connected by an underline character so MANTIS interprets them as one name.)

- **Repeats (V. 14 1).** Entering 14 after the V indicates that you want 14 *additional* vertical occurrences of the field. The 1 indicates you want the repeats to be single-spaced on the screen. (To indicate double-spacing, you would enter 2 in this field.). MANTIS fits as many horizontal repeats as possible into a screen domain before wrapping to the next line. You can repeat numeric fields in both directions; however, text fields can only repeat in one direction. (You can also specify repeats using the Update repeat specifications option from the Screen Design Facility menu).



When specifying field repeats, it is important to stay within the boundaries of your screen design. If you specify more repeats than can fit on your screen design, the fields may overlay each other.

For the CUST_NUMBER field, you will accept the default values for the remaining attributes. However, there are several other attribute fields that you should note for future reference:

- **Length.** Specifies the length of the field. You can alter the length of a text field on this screen to correct a length error in the design.
- **Row/Column.** Indicates the selected field's row/column position.
- **Intensity.** Specifies whether the field should be normal, bright, or hidden. The default value is NORMAL. If, however, you want a bright or hidden field, position the cursor over the N in NOR and enter B (BRIGHT) or H (HIDDEN). Hidden fields are often used for passwords; bright fields are used for headings or key fields.
- **Data Type.** Indicates whether the field will contain text, numeric, or Kanji data. MANTIS automatically supplies the data type for heading fields (HED). Fields with edit characters (that is, dollar sign, decimal point, etc.) are automatically numeric. For all other data fields, the default value is TEXT (TXT). Otherwise, you specify N (NUMERIC), or K (KANJI).
- **Protected.** Specifies whether you want this field protected (that is, read-only capability) or unprotected (that is, available for data entry). Heading fields are automatically protected. The default value is NO. If you want to protect a field, position the cursor over the N in NO and enter Y (YES).
- **Auto Skip.** Indicates that the cursor will skip automatically to the next unprotected data field when a user fills the current field. If autoskip is set to No, the cursor will remain on the field after the user fills it.

For a detailed description of the other available screen design attributes, refer to AD/Advantage MANTIS Facilities OpenVMS/UNIX, P39-1300.

Accept the default values for the other attributes by pressing ENTER. MANTIS highlights the field you just defined.

Supply a symbolic name for the next undefined field. (Call it CUST_NAME.) Enter 14 and 1 for the vertical repeats and displacement values, respectively. Again, accept the defaults for the other attributes by pressing ENTER without modifying the default values:

```
                              B|U|R|R|Y|S
                            CUSTOMER|BROWSE

 CUSTOMER|||||||CUSTOMER|||||||||BRANCH|||||||CREDIT|||||||||CREDIT
  NUMBER|||||||||||NAME|||||||||NUMBER|||||||RATING|||||||||LIMIT

  ######  ####################  ####           ##          $#####




+-------------------------------------------------------------------------------+
! Field Name    : cust_name                      :  Row/Column   : 7   : 11  : !
! Intensity     : NORMAL :      Data Type       : TEXT    :   Auto-Skip   : Y : !
! Reverse/FF    : N : N :       Protected       : N :         Blinking    : N : !
! Underline/FF  : N : N :       Prot Input Only : N :         Uppercase   : N : !
! Extended      : N :           Length          : 20  :       Cursor      : N : !
! Color         :             :  Detectable     : N :         Modified    : N : !
! Left Bar      : N :           Right Bar       : N :         Over Bar    : N : !
! Double Height : N :           Double Width    : N :         Highlight   : N : !
! Vert Rep/Disp : 14  : 1   :   Horiz Rep/Disp  :     :     : Message     : N : !
! Entry Routine :                                 :                           !
!                              Field Sensitive Validation / Forced  : N : N : !
+-------------------------------------------------------------------------------+
```

Provide symbolic names and repeat specification for the next three undefined fields as shown in this table:

Heading	Symbolic name	Repeat specifications
BRANCH NUMBER	CUST_BRCH_NUMBER	V 14 1
CREDIT RATING	CUST_CREDIT_RAT	V 14 1
CREDIT LIMIT	CUST_CREDIT_LIMIT	V 14 1

Accept the default values for the other attributes by pressing ENTER after you supply the name and repeat specifications for each field.

The dollar sign ($) in the CREDIT LIMIT field indicates to MANTIS that this field is masked and numeric (NUM).

Supply the name MESSAGE for the last data field on your screen. The message field appears only once on the screen, so there are no repeat specifications. Move the cursor to the Intensity field and enter a B (for BRIGHT) over the N in NOR. (Bright specifies that the MESSAGE data field will be displayed in high intensity.)

You should also specify that MESSAGE is a protected field, since you do not want the user to be able to alter the information that will appear there. Move the cursor to the Protected field and enter Y (for YES) over the N:

```
                              B|U|R|R|Y|S
                             CUSTOMER|BROWSE

 CUSTOMER|||||||CUSTOMER|||||||||BRANCH|||||||CREDIT|||||||||CREDIT
  NUMBER|||||||||||NAME|||||||||NUMBER|||||||RATING|||||||||LIMIT

   ######  ####################  ####           ##          $#####



+-------------------------------------------------------------------------------+
! Field Name    : message                        :  Row/Column   : 22  : 2   : !
! Intensity     : bright :     Data Type       : TEXT    :   Auto-Skip   : Y : !
! Reverse/FF    : N : N :      Protected       : y :         Blinking    : N : !
! Underline/FF  : N : N :      Prot Input Only : N :         Uppercase   : N : !
! Extended      : N :          Length          : 76  :       Cursor      : N : !
! Color         :           :  Detectable      : N :         Modified    : N : !
! Left Bar      : N :          Right Bar       : N :         Over Bar    : N : !
! Double Height : N :          Double Width    : N :         Highlight   : N : !
! Vert Rep/Disp :     :     :  Horiz Rep/Disp  :     :     : Message     : N : !
! Entry Routine :                                 :                       !
!                              Field Sensitive Validation / Forced  : N : N : !
+-------------------------------------------------------------------------------+

 ################################################################################
```

Accept the default values for the other specifications by pressing ENTER without modifying any other values.

You have now defined all of the data fields for this screen, so press CANCEL to exit from this option and return to the Screen Design Facility menu.

Be sure to press CANCEL only once. If you try to exit to the Facility Selection menu with any unsaved changes, MANTIS will display a warning message and ask for confirmation.

Step 5: Listing field specifications

Now, to verify that you have defined all of the data fields as you intended, select the List field specifications option. (Enter 3 in the action field and press ENTER, or press PF3.) The following screen displays:

```
                          List Field Specifications              Page 1
USERBAME:BURRRYS
Sel---------Field Name----------- Row.Col Len -------Attribute List--------
  "B U R R Y S"                    1,33    11
  "CUSTOMER BROWSE"                2,33    15
  "CUSTOMER       CUSTOMER      " 4,2     64
  "NUMBER           NAME        " 5,3     63
  CUST_NUMBER                      7,3     6
  CUST_NAME                        7,11    20
  CUST_BRCH_NUMBER                 7,33    4
  CUST_CREDIT_RAT                  7,48    2
  CUST_CREDIT_LIMIT / $#####       7,60    6   NUM,MAS









Use <CANCEL> to Exit
```

MANTIS displays a list of the data field names in row/column order, along with the row/column coordinates, the field length, and the attribute specifications. For example, the CUST_NUMBER field appears at row 7, column 3. It has a length of 6 with default field attributes.

MANTIS supplies the MASKED attribute (abbreviated MAS above) for those data fields that contain characters other than #. Since CUST_CREDIT_LIM contains the dollar sign ($), MANTIS assigns the MASKED attribute.

Your attributes should correspond with those in the preceding screen. If they do, press CANCEL to return to the Screen Design Facility menu.

If your attributes do not correspond with those in the preceding screen, press CANCEL to return to the Screen Design Facility menu, then select option 2 (Update field specifications) and correct the fields in error. (For detailed instructions on updating field specifications, see "Step 4: Updating field specifications" on page 43.)

Step 6: Listing repeat specifications

Now, verify that you have defined your repeat specifications as you intended by selecting List repeat specifications. Enter 5 in the action field on the Screen Design Facility menu and press ENTER (or press PF5). MANTIS displays the repeat specification listing:

```
                        List Repeat Specifications                    Page 1

Field Name          Row  Column   Size      Vertical      Horizontal
                                             Repeat  Disp   Repeat  Disp
CUST_NUMBER           7    3       6          14    1
CUST_NAME             7    11      20         14    1
CUST_BRCH_NUMBER      7    33      4          14    1
CUST_CREDIT_RAT       7    48      2          14    1
CUST_CREDIT_LIMI      7    60      6          14    1









Use <RETURN> to Page, <CANCEL> to Exit
```

The MESSAGE field does not display here because it occurs only once on the screen, and so has no repeat specifications.

Your repeat specifications should correspond with those in the preceding screen. If they do, press CANCEL to return to the Screen Design Facility menu.

If your repeat specifications do not correspond with those in the preceding screen, press CANCEL to return to the Screen Design Facility menu, then select option 2 (Update field specifications) and correct the fields in error. (For detailed instructions on updating field specifications, see “Step 4: Updating field specifications” on page 43.)

If you have no changes to make to other attributes, you can also use option 4 (Update repeat specifications) to correct repeat specifications for a screen.

Step 7: Displaying the completed screen design

You can display your final screen design before replacing it in your library by selecting the Display completed design option (number 6 or PF6) from the Screen Design Facility menu. When you do so, the following screen displays:

```
                              B U R R Y S
                           CUSTOMER BROWSE


CUSTOMER          CUSTOMER       BRANCH        CREDIT   CREDIT
 NUMBER             NAME         NUMBER        RATING    LIMIT

 ######   ####################     ####          ##     $#####
 ######   ####################     ####          ##     $#####
 ######   ####################     ####          ##     $#####
 ######   ####################     ####          ##     $#####
 ######   ####################     ####          ##     $#####
 ######   ####################     ####          ##     $#####
 ######   ####################     ####          ##     $#####
 ######   ####################     ####          ##     $#####
 ######   ####################     ####          ##     $#####
 ######   ####################     ####          ##     $#####
 ######   ####################     ####          ##     $#####
 ######   ####################     ####          ##     $#####
 ######   ####################     ####          ##     $#####
 ######   ####################     ####          ##     $#####
 ######   ####################     ####          ##     $#####
################################################################################
```

The screen shows a completed customer browse screen. Note that MANTIS does not display the row scale line or the blank fill characters (|).

If the screen domain is larger than the physical screen, MANTIS automatically initiates window mode. In this case, you can use the window mode PF keys to scroll and view the entire design. (Window mode PF key settings and row/column coordinates display on the last line of the screen.)

Your completed screen design should correspond with the design in the preceding screen. Press CANCEL to return to the Screen Design Facility menu.

Step 8: Saving the screen

You should now save or replace your completed screen in your library. To do so, select the Library functions option from the Screen Design Facility menu. If you have previously saved the screen, the name and description of the screen appear in the relevant fields:

```
SCR002                          M  A  N  T  I  S

                         SCREEN DESIGN LIBRARY FACILITY

     Screen Name ....... : CUST_BROWSE                                        :
     Description ....... : BURRYS CUSTOMER BROWSE SCREEN                     :
     Language .......... : ENGLISH          :
     Screen Size ....... : 24 x 80  : (Rows x Columns)
     Export File Name .. :                                  :

     Save  ...................  1           Sound Alarm ........... : N :
     Replace  ................  2           Full Display .......... : N :
     Fetch  ..................  3           Protect Bottom Line ... : N :
     Delete  .................  4           Field Separators ...... : N :
     Clear Work Area  ........  5           Opaque Map ............ : N :
     Export  .................  6           Automatic Windowing ... : Y :
     Import  .................  7           Horizontal Windowing .. : Y :
                                            Vertical Windowing .... : Y :
     Exit  ..............  CANCEL           Mask Character ........ : # :
                                            Blank Fill Character .. : | :
                : 1  :
```

You can save a screen design only once. After that, you must use the Replace option to store any updates:

- If you *have not* previously saved the screen, enter the symbolic name, and the description as shown. Select the Save option by entering a 1 in the action field and pressing ENTER.
- If you *have* previously saved the screen, select the Replace option by entering a 2 in the action field and pressing ENTER.

MANTIS automatically returns you to the Screen Display Facility menu and displays a confirmation message in the bottom, left corner of the screen.

Step 9: Viewing the directory of screens

Use the Directory of screens option to display an alphabetic listing of all existing screen designs. This list includes the first 16 characters of the screen name and the screen password, format (new or old), and description.

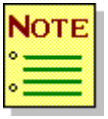
Select the Directory of screens option from the Screen Design Facility menu by entering 8 in the selection field and pressing ENTER, or by pressing PF8. MANTIS displays the directory of screens:

```
   DIR001 BURRYS                          Directory Of Screens          2000/11/30
                                                                       13:06:56
SEL-------------NAME------------- -----------------DESCRIPTION-----------------
   CUST_BROWSE                NEW BURRYS CUSTOMER BROWSE SCREEN
```

Press ENTER to return to the Screen Design Facility menu.

Step 10: Printing the completed screen design

If you would like to print a hard copy of your final screen design, select the Print completed design option from the Screen Design Facility menu. (Enter 9 in the action field and press ENTER, or press PF9.) The printout will be routed to your designated printer.



Your Master User designates your printer in your user profile. If you have no designated printer, MANTIS will display an error message if you try to print your screen.

You can return to the Screen Design Facility menu at any time during the screen design phase and select this option. MANTIS routes the current screen design to your designated printer.

When you select this option, a confirmation message displays to confirm that the screen has printed.

Step 11: Using MANTIS application keys in screen design.

MANTIS terminal I/O is designed for a standard terminal keyboard, consisting of a main keypad, a numeric keypad, cursor movement (arrow) keys, and function keys.

Refer to AD/Advantage MANTIS Facilities OpenVMS/UNIX, P39-1300 and AD/Advantage MANTIS Administration OpenVMS/UNIX, P39-1320, for more detailed information about keyboard interpretation, and how you can effectively customize your keyboard for MANTIS.

MANTIS application keys allow you to design a screen that is larger than the size of your physical terminal.

Numeric keypad key	Function
1	Move the window to the bottom left of the logical display
2	Move the window down by the row increment value
4	Move the window left by the column increment value
6	Move the window to the top left of the logical display
8	Move the window up by the row increment value
9	Move the window to the top right of the logical display

To modify the screen, it must be in your screen design work area. If you have just re-entered the Screen Design Facility from the Facility Selection menu, you must fetch the screen from the library into your work area:

From the Screen Design Facility menu, select Library functions.

Enter the screen name, CUST_BROWSE.

Enter option 3 and press ENTER. MANTIS returns you to the Screen Design Facility menu and displays a confirmation message in the bottom, left corner of the screen.

You can alternately enter CUST_BROWSE in the **Load Screen** field and 1 (Create or Update a Screen) on the option field of the Screen Design Facility menu and press ENTER. This will load the screen and enter the create or update option in one step.

Now, select the Create or update a screen option (1) from the Screen Design Facility menu. Your original design appears:

```
.                                B|U|R|R|Y|S
.                               CUSTOMER|BROWSE
.
.  CUSTOMER|||||||CUSTOMER|||||||||BRANCH|||||||CREDIT|||||||||CREDIT
+   NUMBER|||||||||||NAME|||||||||NUMBER|||||||RATING|||||||||LIMIT
.
.    ######  ####################  ####           ##          $#####
.
.
1
.
.
.
.
.
+
.
.
.
.
2
. ##############################################################################
```

You are going to expand your screen to include a Comments field (length 25) next to the Credit Limit field.

First, expand the size of the logical screen domain by changing the current values for your screen. (Remember that your screen domain defaults to the size of your terminal.)

Press PF12. MANTIS displays the screen domain on the last line of the screen.

```
 .                                B|U|R|R|Y|S
 .                              CUSTOMER|BROWSE
 .
 .  CUSTOMER|||||||CUSTOMER|||||||||BRANCH|||||||CREDIT|||||||||CREDIT
 +   NUMBER|||||||||||NAME|||||||||NUMBER|||||||RATING|||||||||LIMIT
 .
 .   ######  ####################  ####           ##           $#####
 .
 .
 1
 .
 .
 .
 .
 +
 .
 .
 .
 .
 2
Current map domain is: 22  80  : Modify if required and press RETURN
```

From this prompt, you can alter the screen size (row and column domains).

Since you will add a field of 25 bytes, you must extend the screen. The Comments field needs a total of 25 columns. But, because 10 columns are already available on the screen, you only need to extend the screen by 15 columns.

Leave the Row Domain at 22 and change the Column Domain to 95, as shown:

```
.                                B|U|R|R|Y|S
 .                              CUSTOMER|BROWSE
 .
 .  CUSTOMER|||||||CUSTOMER||||||||BRANCH|||||||CREDIT||||||||CREDIT
 +   NUMBER||||||||||NAME|||||||||NUMBER|||||||RATING|||||||||LIMIT
 .
 .   ######  ####################  ####           ##          $#####
 .
 .
 1
 .
 .
 .
 .
 .
 +
 .
 .
 .
 .
 2
Current map domain is: 22  95  : Modify if required and press RETURN
```

Press ENTER. MANTIS redisplays your screen design:

```
 .                                B|U|R|R|Y|S
 .                              CUSTOMER|BROWSE
 .
 .  CUSTOMER|||||||CUSTOMER||||||||BRANCH|||||||CREDIT||||||||CREDIT
 +   NUMBER||||||||||NAME|||||||||NUMBER|||||||RATING|||||||||LIMIT
 .
 .   ######  ####################  ####           ##          $#####
 .
 .
 1
 .
 .
 .
 .
 .
 +
 .
 .
 .
 .
 2
 . ###############################################################################
```

Press 6 on the numeric keypad to move your terminal window to the right:

```
.              B|U|R|R|Y|S
.           CUSTOMER|BROWSE
.
. OMER|||||||BRANCH|||||||CREDIT|||||||CREDIT
+ ME|||||||||NUMBER|||||||RATING|||||||||LIMIT
.
. #########   ####           ##          $#####
.
.
1
.
.
.
.
.
+
.
.
.
.
2
. ############################################################
```

Remember, MANTIS reserves the first column on the screen for the row scale line. The last two lines are reserved for the message line, row/column coordinates, command line, and the key simulation field. These fields remain on your screen even when you move your window.

With your cursor positioned anywhere on the same line as “BURRYS,” press PF3 to display the column scale line. It will overlay the top line of your screen, making it temporarily invisible:

```
. ....+....3....+....4....+....5....+....6....+....7....+....8....+....9....+
.           CUSTOMER|BROWSE
.
. OMER|||||||BRANCH|||||||CREDIT|||||||CREDIT
+ ME|||||||||NUMBER|||||||RATING|||||||||LIMIT
.
. #########   ####           ##          $#####
.
.
1
.
.
.
.
.
+
.
.
.
.
2
. ############################################################
```

Later, you will remove the column scale line, and the top line of your screen will display again.

The current column position within the work area is indicated by the column scale line. This scale line is adjusted as your terminal window moves across the work area.

With your cursor next to the word “LIMIT” (in the CREDIT LIMIT heading), enter vertical bars up to position 70. With your cursor at column 70, add the heading and the 25 hash signs for the data characters of the Comments field:

```
. ....+....3....+....4....+....5....+....6....+....7....+....8....+....9....+
.           CUSTOMER|BROWSE
.
. OMER|||||||BRANCH|||||||CREDIT|||||||CREDIT
+ ME|||||||||NUMBER|||||||RATING|||||||||LIMIT||||--------COMMENTS---------
.
. #########   ####          ##          $#####    #########################
.
.
1
.
.
.
.
+
.
.
.
.
2
. ############################################################
```

Press ENTER to save the screen modifications in your work area.

When you are finished entering the Comments field, position the cursor anywhere on the column scale line and press PF3. The column scale line is removed from your screen, and the top line of your screen is visible again:

```
.             B|U|R|R|Y|S
.           CUSTOMER|BROWSE
.
. OMER|||||||BRANCH|||||||CREDIT|||||||CREDIT
+ ME|||||||||NUMBER|||||||RATING|||||||||LIMIT||||--------COMMENTS---------
.
. #########   ####           ##          $#####    ########################
.
.
1
.
.
.
.
+
.
.
.
.
2
. ############################################################
```

Press 4 on the numeric keypad to scroll your terminal window back to the left:

```
.                                B|U|R|R|Y|S
.                              CUSTOMER|BROWSE
.
.  CUSTOMER|||||||CUSTOMER|||||||BRANCH|||||||CREDIT|||||||CREDIT
+   NUMBER|||||||||||NAME|||||||||NUMBER|||||||RATING|||||||||LIMIT||||--------C
.
.   ######  ####################  ####           ##          $#####    #########
.
.
1
.
.
.
.
+
.
.
.
.
2
. ###############################################################################
```

You could also have pressed 7 on the numeric keypad to return the screen to its origin (row 1, column 1).

Now, you need to specify the field attributes for the Comments field that you just added to your screen design. Press CANCEL to return to the Screen Design Facility menu, then select the Update field specifications option.

MANTIS displays your screen design. Press PF3 to display the undefined fields (in this case, the Comments field). The Comments field is highlighted and a boxed prompt lists the current specifications for that field:

```
 .             B|U|R|R|Y|S
 .           CUSTOMER|BROWSE

   OMER|||||||BRANCH|||||||CREDIT||||||||CREDIT
 + ME|||||||||NUMBER|||||||RATING|||||||||LIMIT||||--------COMMENTS---------
 .
 . #########   ####           ##          $#####    ##########################
 .
 .

+-------------------------------------------------------------------------------+
! Field Name    :                                  :  Row/Column   :  7  : 70  : !
! Intensity     : NORMAL :     Data Type       : TEXT    :   Auto-Skip   : Y : !
! Reverse/FF    : N : N :      Protected       : N :         Blinking    : N : !
! Underline/FF  : N : N :      Prot Input Only : N :         Uppercase   : N : !
! Extended      : N :          Length          : 25  :       Cursor      : N : !
! Color         :           :  Detectable      : N :         Modified    : N : !
! Left Bar      : N :          Right Bar       : N :         Over Bar    : N : !
! Double Height : N :          Double Width    : N :         Highlight   : N : !
! Vert Rep/Disp :     :     :  Horiz Rep/Disp  :     :     : Message     : N : !
! Entry Routine :                                 :                           !
!                              Field Sensitive Validation / Forced  : N : N : !
+-------------------------------------------------------------------------------+
 .############################################################
```

If a selected field is outside the boundary of the current window display, MANTIS moves the window so you can view the field that you are updating.

Enter CUST_COMMENTS for the field name and specify the necessary vertical repeats. To do so, move the cursor to Repeats. After the V (vertical), enter 14 for the number of vertical repeats, and 1 for the vertical displacement:

```
 .             B|U|R|R|Y|S
 .           CUSTOMER|BROWSE

   OMER|||||||BRANCH|||||||CREDIT|||||||CREDIT
 + ME||||||||||NUMBER|||||||RATING|||||||||LIMIT||||--------COMMENTS---------
 .
 . #########   ####           ##          $#####    ##########################
 .
 .
+-------------------------------------------------------------------------------+
! Field Name    : cust_comments                 :  Row/Column   :  7  : 70  : !
! Intensity     : NORMAL :     Data Type       : TEXT    :   Auto-Skip   : Y : !
! Reverse/FF    : N : N :      Protected       : N :         Blinking    : N : !
! Underline/FF  : N : N :      Prot Input Only : N :         Uppercase   : N : !
! Extended      : N :          Length          : 25  :       Cursor      : N : !
! Color         :           :  Detectable      : N :         Modified    : N : !
! Left Bar      : N :          Right Bar       : N :         Over Bar    : N : !
! Double Height : N :          Double Width    : N :         Highlight   : N : !
! Vert Rep/Disp :  14 :  1  :  Horiz Rep/Disp  :     :     : Message     : N : !
! Entry Routine :                                :                           !
!                             Field Sensitive Validation / Forced  : N : N : !
+-------------------------------------------------------------------------------+
 .#############################################################
 ASP030A: Modify the settings by overtyping and press 'ENTER'
```

Press ENTER.

Remember, MANTIS supplies default values for all the other attributes if you do not specify any new values.

Press CANCEL to return to the Screen Design Facility menu.

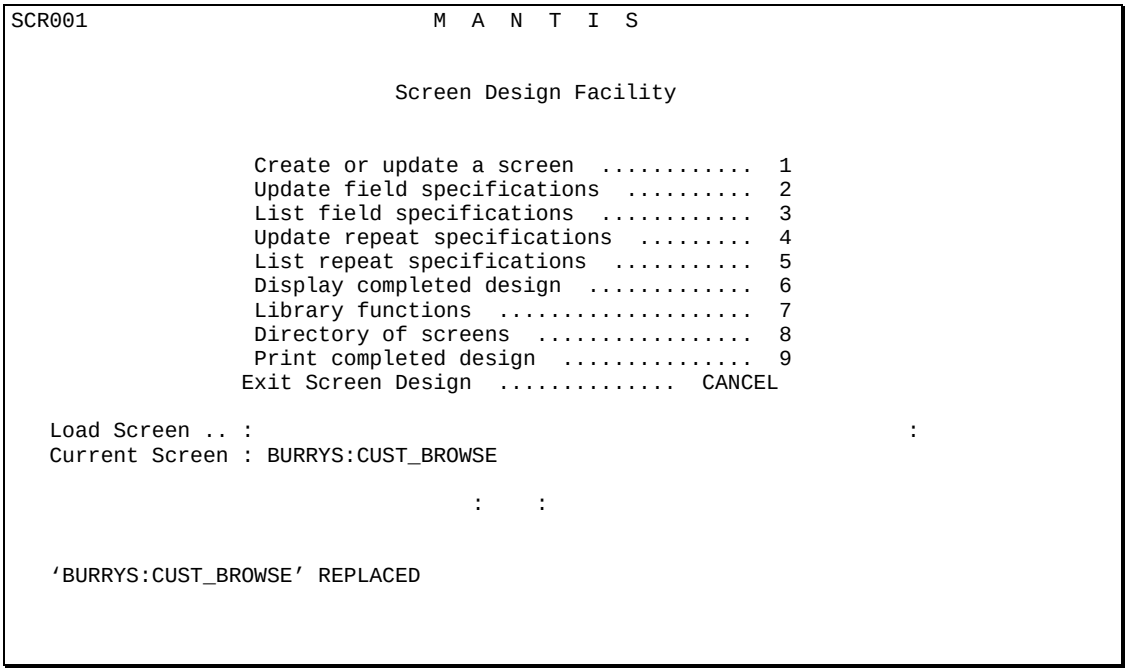
Save your modifications by selecting the Library functions option. The screen name and description are provided automatically by MANTIS as shown:

```
SCR002                          M  A  N  T  I  S

                         SCREEN DESIGN LIBRARY FACILITY

     Screen Name ....... : BURRYS:CUST_BROWSE                              :
     Description ....... : BURRYS CUSTOMER BORWSE SCREEN                   :
     Language .......... : ENGLISH         :
     Screen Size ....... : 24 x 80  : (Rows x Columns)
     Export File Name .. :                                      :

     Save  ...................  1          Sound Alarm ........... : N :
     Replace  ................  2          Full Display .......... : N :
     Fetch  ..................  3          Protect Bottom Line ... : N :
     Delete  .................  4          Field Separators ...... : Y :
     Clear Work Area  ........  5          Opaque Map ............ : N :
     Export  .................  6          Automatic Windowing ... : N :
     Import  .................  7          Horizontal Windowing .. : Y :
                                           Vertical Windowing .... : Y :
     Exit  ...............  CANCEL         Mask Character ........ : # :
                                           Blank Fill Character .. : | :
               :    :
```

Select the Replace option (enter 2 and press ENTER, or press PF2).

Notice that the value for the physical column is still 80, even though the domain of the logical display is 95.

MANTIS returns you to the Screen Design Facility menu and displays a confirmation message in the bottom, left corner of the screen:



To display your updated screen, select the Display completed design option from the Screen Design Facility menu.

MANTIS automatically places you in screen display window mode because you increased the screen domain beyond the boundaries of the physical screen.

```
                                B U R R Y S
                             CUSTOMER BROWSE


CUSTOMER            CUSTOMER        BRANCH        CREDIT    CREDIT
 NUMBER               NAME          NUMBER        RATING     LIMIT     --------COM

 ######    ####################      ####           ##      $#####     ##########
 ######    ####################      ####           ##      $#####     ##########
 ######    ####################      ####           ##      $#####     ##########
 ######    ####################      ####           ##      $#####     ##########
 ######    ####################      ####           ##      $#####     ##########
 ######    ####################      ####           ##      $#####     ##########
 ######    ####################      ####           ##      $#####     ##########
 ######    ####################      ####           ##      $#####     ##########
 ######    ####################      ####           ##      $#####     ##########
 ######    ####################      ####           ##      $#####     ##########
 ######    ####################      ####           ##      $#####     ##########
 ######    ####################      ####           ##      $#####     ##########
 ######    ####################      ####           ##      $#####     ##########
 ######    ####################      ####           ##      $#####     ##########
 ######    ####################      ####           ##      $#####     ##########
 ######    ####################      ####           ##      $#####     ##########
###############################################################################
```

You can use the numeric keypad keys to scroll around the screen display. You can also use GOLD/W to display the row/column coordinates at the bottom, right corner of the screen. They specify the upper left corner of the physical screen's position on the work area.

You can also scroll around the screen by changing the values of the row/column coordinates. To do so, enter the new position for the top left corner of the logical screen and press ENTER.

Press 6 on the numeric keypad to scroll your terminal screen to the right. The entire Comments field is now visible:

```
            B U R R Y'S
          CUSTOMER BROWSE


USTOMER       BRANCH        CREDIT   CREDIT
 NAME         NUMBER        RATING    LIMIT    --------COMMENTS---------

##########     ####           ##     $#####    ########################
##########     ####           ##     $#####    ########################
##########     ####           ##     $#####    ########################
##########     ####           ##     $#####    ########################
##########     ####           ##     $#####    ########################
##########     ####           ##     $#####    ########################
##########     ####           ##     $#####    ########################
##########     ####           ##     $#####    ########################
##########     ####           ##     $#####    ########################
##########     ####           ##     $#####    ########################
##########     ####           ##     $#####    ########################
##########     ####           ##     $#####    ########################
##########     ####           ##     $#####    ########################
##########     ####           ##     $#####    ########################
##########     ####           ##     $#####    ########################
#########################################################
```

To return the screen to the origin, press 4 on the numeric keypad:

```
                              B U R R Y S
                            CUSTOMER BROWSE


CUSTOMER           CUSTOMER       BRANCH        CREDIT   CREDIT
 NUMBER              NAME         NUMBER        RATING    LIMIT    --------COM

 ######   ####################     ####          ##     $#####    ###########
 ######   ####################     ####          ##     $#####    ###########
 ######   ####################     ####          ##     $#####    ###########
 ######   ####################     ####          ##     $#####    ###########
 ######   ####################     ####          ##     $#####    ###########
 ######   ####################     ####          ##     $#####    ###########
 ######   ####################     ####          ##     $#####    ###########
 ######   ####################     ####          ##     $#####    ###########
 ######   ####################     ####          ##     $#####    ###########
 ######   ####################     ####          ##     $#####    ###########
 ######   ####################     ####          ##     $#####    ###########
 ######   ####################     ####          ##     $#####    ###########
 ######   ####################     ####          ##     $#####    ###########
 ######   ####################     ####          ##     $#####    ###########
 ######   ####################     ####          ##     $#####    ###########
#############################################################################
```

Now, press ENTER or CANCEL to return to the Screen Design Facilities menu and proceed with the exercises in the next section.

Exercises

Three additional screens are needed for the Burrys application:

- Burrys customer accounts menu (CUST_MENU)
- Burrys new customer entry screen (CUST_ENTRY)
- State code entry screen (STATE_CODE)

Create these screens using the procedures outlined in this chapter and the data supplied with each exercise.

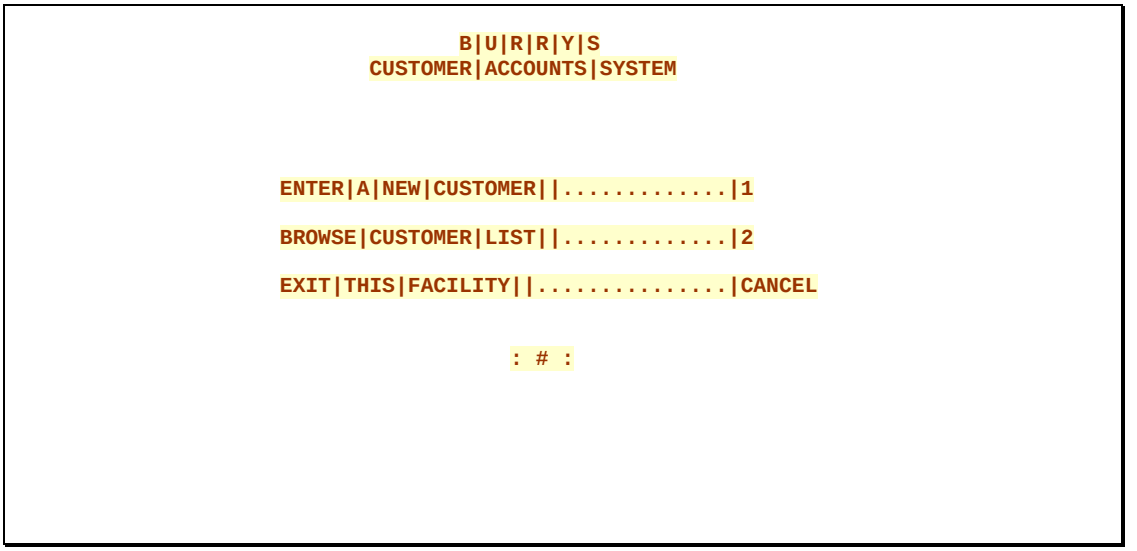
To begin:

1. Press CANCEL to return to the Facility Selection menu from the Screen Design Facility menu. This clears the Burrys customer browse screen from your work area. (Remember, your design has been permanently saved in your library.)
2. Select the Design a Screen option. When the Screen Design Facility menu appears, you are ready to begin the first exercise.

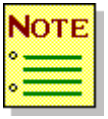
Our versions of the completed screens appear in “Exercise examples” on page 189.

Exercise 1: Creating the customer menu screen

Create the customer accounts menu to correspond with the following screen design:



Notice that a space appears before and after the hash sign (#). At least one space must separate fields (except when you use the blank fill character as a non-displaying character for heading fields).



Remember to press ENTER to save your screen design temporarily.

Press CANCEL to return to the Screen Design Facility menu; then, proceed with Update field specifications. (For step-by-step directions for adding field specifications, see “Step 4: Updating field specifications” on page 43.)

The symbolic name (supplied next to Field Name) for the one-byte date field (between the colons) should be ACTION. The field has a data type of NUMERIC:

```
+-------------------------------------------------------------------------+
! Field Name    : action                        :  Row/Column   : 18  : 37  : !
! Intensity     : NORMAL :     Data Type       : num  :   Auto-Skip   : Y : !
! Reverse/FF    : N : N :      Protected       : N :         Blinking    : N : !
! Underline/FF  : N : N :      Prot Input Only : N :         Uppercase   : N : !
! Extended      : N :          Length          : 1   :       Cursor      : N : !
! Color         :           :  Detectable      : N :         Modified    : N : !
! Left Bar      : N :          Right Bar       : N :         Over Bar    : N : !
! Double Height : N :          Double Width    : N :         Highlight   : N : !
! Vert Rep/Disp :     :     :  Horiz Rep/Disp  :     :     : Message     : N : !
! Entry Routine :                              :                         !
!                             Field Sensitive Validation / Forced  : N : N : !
+-------------------------------------------------------------------------+
                   VIEW|CUSTOMER|BROWSES|SCREEN  .....|2

                   EXIT|THIS|FACILITY  ..............|CANCEL


                                    : # :
```

Press ENTER to update the field specifications.

Press CANCEL, and MANTIS returns you to the Screen Design Facility menu. Proceed with the sequence of functions listed on the menu.

This screen needs no repeat specifications.

Save your design (via library functions), specifying a name of CUST_MENU, a password, and the description “BURRYS CUSTOMER ACCOUNTS MENU.”

Exercise 2: Creating the new customer entry screen

Create the new customer entry screen to correspond with the following screen design:

```
                    B|U|R|R|Y|S
                NEW|CUSTOMER|ENTRY


          NAME:                      ####################
          ADDRESS:                   ####################
          CITY:                      #############
          STATE:                     ##
          ZIP|CODE:                  #####

          CUSTOMER|NUMBER:           ######
          CLASS:                     ##
          CUSTOMER|CREDIT|RATING:    ##
          CREDIT|LIMIT:              #####
          BRANCH|NUMBER:             ####
          COMMENTS:                  ########################

################################################################################
```

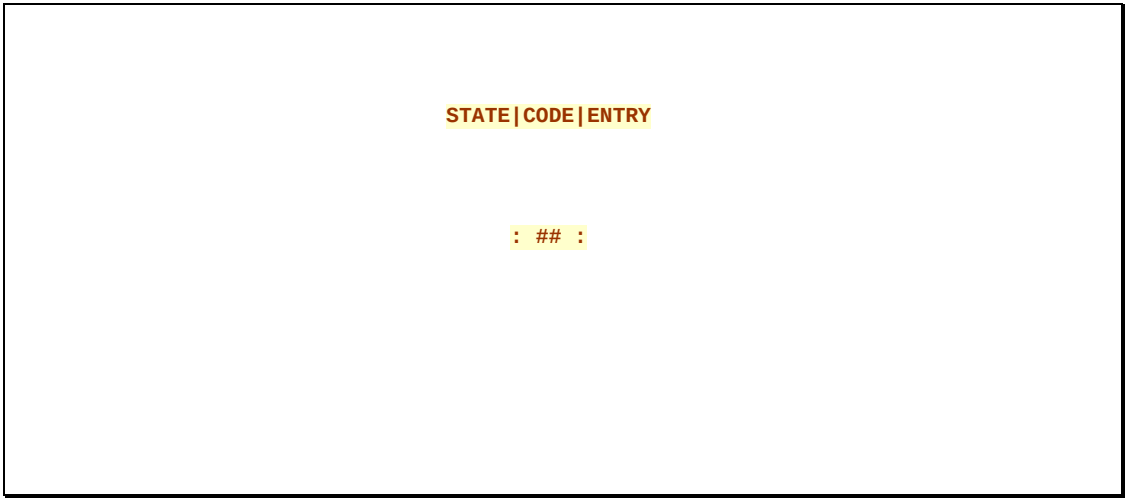
Finish the screen design process using the following data specifications:

Heading	Symbolic name	Field length	Attributes
NAME	CUST_NAME	20	Autoskip
ADDRESS	CUST_ADDRESS	20	Autoskip
CITY	CUST_CITY	13	Autoskip
STATE	CUST_STATE	2	Autoskip
ZIP CODE	CUST_ZIP_CODE	5	Numeric, autoskip
CUSTOMER NUMBER	CUST_NUMBER	6	Autoskip
CLASS	CUST_CLASS	2	Autoskip
CUSTOMER CREDIT RATING	CUST_CREDIT_RAT	2	Autoskip
CREDIT LIMIT	CUST_CREDIT_LIM	5	Numeric, autoskip
BRANCH NUMBER	CUST_BRCH_NUMBER	4	Autoskip
COMMENTS	CUST_COMMENTS	25	Autoskip
	MESSAGE	76	Bright, protected

When you are finished creating your screen, save your design using the library functions. Specify a name of CUST_ENTRY, a password, and the description “BURRYS NEW CUSTOMER ENTRY SCREEN.”

Exercise 3. Creating the state code entry screen

Create the state code entry screen to correspond with the following screen design:



Proceed with the rest of the screen design process as presented on the Screen Design Facility menu. The symbolic name for the two-byte data field should be CUST_STATE.

Save your design via library functions. Specify the name STATE_CODE and supply the description “STATE CODE ENTRY SCREEN.”

Exercise 4: Viewing the directory of screens

When you have finished designing and saving these screens, select the Directory of screens option from the Screen Design Facility menu. (Enter 8 and press ENTER, or press PF8.) Your Directory of Screens should correspond with the following screen:

```
   DIR001 EXAMPLES                    Directory Of Screens             2000/11/30
                                                                       14:37:14
SEL-------------NAME------------ -----------------DESCRIPTION------------------
   CUST_BROWSE                BURRYS CUSTOMER BROWSE SCREEN
   CUST_ENTRY                 BURRYS NEW CUSTOMER ENTRY SCREEN
   CUST_MENU                  BURRYS CUSTOMER ACCOUNTS MENU
   STATE_CODE                 STATE CODE ENTRY SCREEN
```

Press CANCEL to return to the Screen Design Facility menu.

You have now completed the screen design portion of the Burrys scenario. To continue with the tutorial, proceed to “Creating a MANTIS file” on page 77.

3

Creating a MANTIS file

In this chapter you will create the Burrys customer information file, which is one of two files used in the Burrys scenario. When you create MANTIS programs later in this tutorial, the records in this file will be read and displayed on the Burrys customer browse screen (CUST_BROWSE) that you designed in chapter 2.

Learning outline

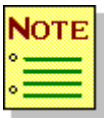
In this chapter you will learn how to:

- Specify a name and define the access levels for a file
- Create or update the record layout for a file, defining the fields it contains and the format in which MANTIS stores the data
- Save a new file, or changes to an existing file, using the File Design Library Facility
- View the directory of files
- Print the completed file design

Basic concepts: Understanding MANTIS file design

Use the MANTIS File Design Facility to create, save, update, and maintain file views for internal MANTIS files. A *file view* contains detailed information about the contents and format of the data stored in a file, and allows you to control access to a file by password-protecting certain functions performed on the file data.

Use the Update record layout option to create or update the record layout for a file. A *record layout* defines the format in which MANTIS stores and transmits file data. The record layout for a file can contain up to 64 fields. (Each page of the record layout displays 16 fields.)



In our application, the element names in the MANTIS record layout definition must be exactly the same as the symbolic name of the corresponding data field in your screen design.

Each element receives a data type of TEXT, BIG, SMALL, or KANJI. For TEXT and KANJI items, the length must be specified in the Dimensions field. You do not specify a length for SMALL and BIG elements.

Elements can have these attributes:

- **KEY.** Records in the file are ordered and accessed by the key element. Each record must have at least one key element, and that must be the first element in the record. The key or concatenated keys can be up to 32 bytes in length.
- **SCRAMBLE.** Indicates that the data in the field will be stored in an encrypted format in the file, for security purposes.

Step-by-step: Creating the customer information file

Now that you understand some basic concepts for MANTIS file design, you're ready to design your first MANTIS file.

To begin, select the Design a File option from the MANTIS Facility Selection menu. The File Design Facility menu displays:

```
MFV001                          M  A  N  T  I  S

                              File Design Facility




                    Create or update file profiles  .......  1
                    Update record layout  .................  2
                    Library functions  ....................  3
                    Directory of file profiles  ...........  4
                    Print completed design  ...............  5
                    Exit File Design  .................  CANCEL


    Load File .. :                                                  :
    Current File :

                                       :   :
```

The File Design Facility menu presents the options in the order you perform them when you create a new MANTIS file view. If you are updating a file view, you must first select the Library functions option to fetch the file view from your library, then update the information as desired.

You can move among the File Design Facility menu options without losing the file view design currently in your work area.

You must use the library functions to save your new or updated file before exiting from the File Design Facility. If you attempt to exit from the facility without saving your changes, MANTIS asks you to confirm your exit. If you exit without saving, you lose your file design (if it is new), or any changes you have made since you last saved or replaced the design.

Step 1: Creating the file

Since you are designing a new file, you will follow the sequence of functions as they are listed on the File Design Facility menu (that is, first create a file profile, then update the record layout, etc.). In this step, you will specify the general characteristics for the file.

Select the Create or update file profiles option by entering 1 in the action field and pressing ENTER. The File Design Facility screen displays, with the cursor in the Name and description of file field. Enter the file information as it appears on the following screen:

```
MFV002                          M  A  N  T  I  S

                              File Design Facility



          Name and description of file  ..... : cust_info        :
          : burrys customer information file                      :
          Associated record layout :                              :
          Password for viewing  ................ :                :
          Password for altering  ............... :                :
          Password for deleting/inserting  ..... : password       :
          Status  ............................. : active         :



          Last profile update date  ............ : YYYY/MM/DD :
          Last profile update time  ............ :          :

          Field Count   ........................ :          :
```

A file name can contain 1-16 alphanumeric characters, and a file description can contain 1-32 alphanumeric characters. The description helps you to identify the file in your file directory.

Do not supply an entry for Associated record layout. (You would supply an associated record layout name if you were using an identical record layout from another file, but in this case you will be creating a new record layout.)

It is not necessary to specify a password for each of the three password levels unless different passwords are desired for security reasons. (The Deleting/inserting specification includes privileges for altering and viewing, and the Altering specification includes privileges for viewing.)

Enter ACTIVE in the Status field. Entering anything other than ACTIVE (for example, ACTIVE␣ or OBSOLETE) prevents a program from accessing the file.

MANTIS maintains the last three fields (Date of Last Profile Update through Field Count) on the File Design Facility screen.

When you have completed the entries as shown in the previous screen, press ENTER. MANTIS accepts your entries and automatically returns you to the File Design Facility menu:

```
MFV001                          M  A  N  T  I  S


                             File Design Facility




                   Create or update file profiles  .......  1
                   Update record layout  .................  2
                   Library functions  ....................  3
                   Directory of file profiles  ...........  4
                   Print completed design  ...............  5
                   Exit File Design  ................  CANCEL


    Load File .. :                                                :
    Current File : CUST_INFO

                                     :   :
```

Step 2: Updating the record layout

Now you must create a record layout for the file, to specify the format in which MANTIS stores and transmits file data. The record layout for a file can contain up to 64 fields, and each page of the record layout displays 16 fields.

Select the Update record layout option from the File Design Facility menu by entering 2 in the action field and pressing ENTER. The MANTIS Record Layout Definition screen displays, with the cursor positioned in the Page field:

```
MFV004 Page No : 1 :            Update Record Layout              2000/11/30
                                                                   15:00:30
  Element Count :                                                Size : 32
  ELEM  -------------NAME-------------  TYPE   DIMENSIONS  ----ATTRIBUTES----

     _   _                         _          _     _     _




              PF1-PF4 Page; CANCEL Exit
```

MANTIS supplies the Date, Time, Page, Element Count, and Size fields.

The underline characters in the preceding screen represent the built-in tabs for this option. Use the Tab key to move from field to field and enter the specified data at the tab positions.

When entering or updating an element in the record layout, you must begin each line with an A, I, or D to indicate the action you want to perform (for Alter, Insert, or Delete respectively).

Begin by entering the specifications for the first element as they appear in the following screen:

```
MFV004 Page No : 1 :          Update Record Layout              2000/11/30
                                                                  15:00:30
  Element Count :                                              Size : 32
     Element  ------Name------  Data-type  Dimensions  ----Attributes----

     i   1    cust_number       t           6           k









                PF1-PF4 Page; CANCEL Exit
```

To enter the specifications for the first element:

1. Use the Tab key to move the cursor from the Page field to the first tab position on line 1.
2. Enter an i to indicate that you are inserting a new line. The cursor automatically moves to the next tab position.
3. Enter 1 to indicate the line number; then, press Tab to move to the Name field.
4. Enter cust_number in the Name field and press Tab again to move to the Data-type field.

5. You can specify TEXT, BIG, SMALL, or KANJI (T, B, S, or K) for the data type of each element. When selecting the data type, use:

   - TEXT for all alphanumeric fields (requires you to enter a length for the field under Dimensions)

   - BIG for a numeric field of up to 14 significant digits (recommended when using decimals)

   - SMALL for a numeric field of up to 6 significant digits (normally, an integer field)

   - KANJI for a Kanji data field (for Asian language support; requires you to enter a length for the field under Dimensions)

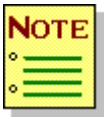
   Since CUST_NUMBER is a text field, enter T in the Data-type field; then, press Tab to move to the Dimensions field.

6. Enter 6 in the Dimensions field; then, press Tab twice to move to the Attribute field. (Use the second tab in the Dimensions field only when you are defining a list of fields, or an array.)

7. Enter K in the Attribute field to indicate that CUST_NUMBER is the key to this file (that is, records in this file are ordered and accessed by CUST_NUMBER). The key field(s) must be defined first in the list of fields.

Enter the remaining elements of the record layout as they appear in the following screen:

```
MFV004 Page No : 1 :          Update Record Layout              2000/11/30
                                                                  15:00:30
  Element Count :                                              Size : 32
     Element  ------Name------  Data-type  Dimensions  ----Attributes----

     i   1    cust_number       t           6           k
     i   2    cust_name         t           20
     i   3    cust_address      t           20
     i   4    cust_city         t           13
     i   5    cust_state        t           2
     i   6    cust_zip_code     s
     i   7    cust_class        t           2
     i   8    cust_credit_rat   t           2
     i   9    cust_credit_lim   s                       s
     i   10   cust_comments      t           25
     i   11   cust_brch_number   t           4




              PF1-PF4 Page; CANCEL Exit
```



The element names that you supply here must be identical to the symbolic names you gave to the corresponding data fields during screen design. Using identical symbolic names will enable data to be passed directly from this file to data fields on the customer browse screen without extra programming.

Notice that you supply dimensions for the text fields only. (Using a data type of BIG or SMALL automatically specifies the length of numeric fields.)

For CUST_CREDIT_LIM, supply the attribute S (SCRAMBLE) to indicate that the data in this field will be stored in an encrypted format.

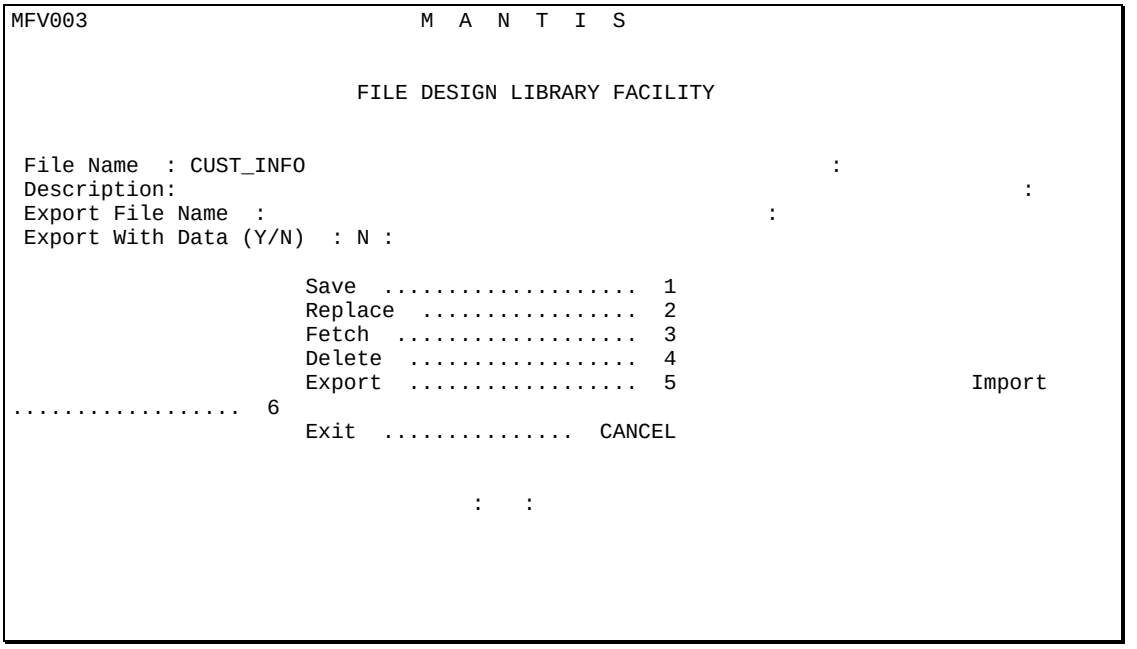
After you have entered all of the data, press ENTER. MANTIS temporarily stores the data in your work area and updates the element count and size:

```
MFV004 Page No : 1 :           Update Record Layout              2000/11/30
                                                                   15:00:30
  Element Count : 11                                          Size : 138

     Element  ------Name------  Data-type  Dimensions  ----Attributes----

         1    CUST_NUMBER       TEXT       6           KEY
         2    CUST_NAME         TEXT       20
         3    CUST_ADDRESS      TEXT       20
         4    CUST_CITY         TEXT       13
         5    CUST_STATE        TEXT       2
         6    CUST_ZIP_CODE     SMALL
         7    CUST_CLASS        TEXT       2
         8    CUST_CREDIT_RAT   TEXT       2
         9    CUST_CREDIT_LIM   SMALL                  SCRAMBLE
         10   CUST_COMMENTS     TEXT       25
         11   CUST_BRCH_NUMBER  TEXT       4




              PF1-PF4 Page; CANCEL Exit
```

Press CANCEL to return to the File Design Facility menu.

Step 3: Saving the file

To save the file, select Library functions by entering 3 in the action field and pressing ENTER. The File Design Library Facility displays:



MANTIS displays the name of the file in the Name field, so you do not need to enter it. To save the file, select the Save option (enter 1 and press ENTER, or press PF1).

MANTIS automatically returns you to the File Design Facility menu and displays a confirmation message in the lower, left corner of the screen.

Step 4: Viewing the directory of files

Use the Directory of file profiles option to display an alphabetic listing of all existing files. To display the list, select this option from the File Design Facility menu by entering 4 in the selection field and pressing ENTER (or by pressing PF4). MANTIS displays the Directory of MANTIS Files screen:

```
DIR001 BURRYS             Directory of Files                YYYY/MM/DD
                                                              HH:MM:SS
------Name------ -----Status----- -------------Description-------------------
CUST_INFO        ACTIVE           BURRYS CUSTOMER INFORMATION FILE
```

If your directory has entries filling more than one page, you can page through them by pressing ENTER.

Press ENTER to return to the File Design Facility menu.

Step 5: Printing the completed file design

If you would like to view a printed copy of the file design, select the Print completed design option by entering 5 in the action field and pressing ENTER. The printout will be routed to your designated printer.

Exercise

In this chapter, you created a new file, the Burrys customer information file, by following the order of functions presented on the File Design Facility menu. One other file is needed for the Burrys scenario: the state codes file.

Use CANCEL to exit to the Facility Selection menu in order to clear your work area. Again, select the Design a File option. Create the state codes file by following the same procedures that you used to create the previous file, but supply the following data on the File Design Facility screen:

Field	Data
File name	STATE_CODES
Description	State codes file
Password	PASSWORD
Status	ACTIVE

The record layout contains one element: CUST_STATE. It is a text element with a dimension of 2. It is also the key element. The final file design should look like this:

```
MFV004 Page No : 1 :            Update Record Layout              2000/11/30
          STATE_CODES                                                15:16:21
  Element Count : 1                                        Size : 32
     Element  ------Name------  Data-type  Dimensions  ----Attributes----

         1    CUST_STATE        TEXT       2           KEY









               PF1-PF4 Page; CANCEL Exit
```

After you design and save the file, check the Directory of MANTIS Files:

```
   DIR001 BURRYS                      Directory Of Files          2000/11/30
                                                                    15:18:36
SEL-------------NAME------------- -----------------DESCRIPTION-----------------
   CUST_INFO                      BURRYS CUSTOMER INFORMATION FILE
   STATE_CODES                    STATE CODES FILE
```

Press CANCEL to return to the File Design Facility menu. Press CANCEL again to exit to the MANTIS Facility Selection menu. Then, proceed with “Creating a prompte” on page 91.

4

Creating a prompter

In this chapter, you will design the state codes prompter for the Burrys scenario, which displays the two-letter abbreviation for each state. You will use this prompter with the customer entry program that you will create later in the tutorial.

Learning outline

In this chapter you will learn how to:

- Accept default tab values, or set new tabs for prompter design
- Change the prompter design default tab character
- Enter, update, and delete prompter text
- Save a new prompter, or changes to an existing prompter, using the Prompter Design Library Facility
- View the directory of prompters
- View the completed prompter design
- Print the completed prompter design

Basic concepts: Understanding prompter design

You can use prompters to present online help information to users. This includes online help for programs, files, and screens. You can also use prompters to document certain aspects of company policy or procedure.

The data on prompter screens is static, and remains the same except for occasional updates.

A MANTIS prompter allows up to 80 lines of text. Each screen displays 20 lines of the prompter, so you can have up to four pages of text per prompter. If this is not enough space, you can add pages by chaining to another prompter (specified in the Chain to next prompter field in library functions).

If your terminal provides lowercase support for text characters, MANTIS accepts the text you enter in a prompter as you enter it (any combination of uppercase and lowercase characters). Check with your Master User to find out if this capability is available in your environment.

If your terminal supports lowercase, you can enter the description and password for a prompter using lowercase characters at the Prompter Design Facility menu. The description entered here becomes the title of the prompter.

Step-by-step: Creating the state codes prompter

Now that you understand some basic concepts of MANTIS prompter design, you're ready to design your first MANTIS prompter.

To begin, select the Design a Prompter option from the MANTIS Facility Selection menu. The Prompter Design Facility menu displays:

```
PRD001                          M  A  N  T  I  S


                           Prompter Design Facility




                    Create or update a prompter  .........  1
                    Set tabs  ............................  2
                    Library functions  ...................  3
                    Directory of prompters  ..............  4
                    Display completed design  ............  5
                    Print completed design  ..............  6
                    Exit Prompter Design  ............  CANCEL


  Load Prompter .. :                                                :
  Current Prompter :                                                :
                                    :   :
```

The Prompter Design Facility menu presents the options in the order in which you perform them when you create a new MANTIS prompter. (If you are updating an existing prompter, you must first select the Library functions option to fetch the prompter from your library, then update the information as desired.)

You can move among the Prompter Design Facility menu options without losing the prompter design currently in your work area.

You must use the library functions to save the new or updated prompter before you exit from the Prompter Design Facility. If you attempt to exit from the facility without saving current changes, MANTIS asks you to confirm your exit. If you exit without saving, you lose your prompter (if it is new), or any changes you have made since you last saved the prompter.

Step 1: Setting the tabs for prompter design

When you are designing a new prompter, you can follow the sequence of functions as they appear on the Prompter Design Facility menu, with one exception: if you want to change the default tab character and positions, you must use the Set tabs option first.

Tabs make it easy for you to format a prompter. You simply enter each line of text, indicating a tab character when you want MANTIS to skip to the next tab position. When you press ENTER, MANTIS moves the text to the indicated position(s). (You will see how the tab settings work as you create the state codes prompter.)

The scale line across the top of the prompter design work area reflects the tab settings specified for the prompter design currently in the work area. This screen illustrates the default tab settings for prompter design:

```
PRD003  Page = 1               Create Or Update A Prompter        Lines =0
   ....|.@..1....|.@..2....|.@..3....|.@..4....|.@..5....|.@..6....|.@..7....|..
```

The at sign (@) is the default tab character, and it displays in the default tab positions at columns 7, 17, 27, 37, 47, 57 and 67.

The default tab positions will not work for the state codes prompter, since it requires a three-column format. Therefore, you must begin by selecting the Set tabs option from the Prompter Design Facility menu.

You can use the Set tabs option to:

- Change the default tab character
- Change the default tab positions to reflect the format you need for your prompter design

Select the Set tabs option from the Prompter Design Facility menu by entering 2 in the action field and pressing ENTER. MANTIS displays the following screen:

```
PRD006                          M  A  N  T  I  S


                          Prompter Design Facility




                  Tab character  ..................... @

                  Tab positions  ..................... 7
                                                      17
                                                      27
                                                      37
                                                      47
                                                      57
                                                      67
```

Notice that the default positions are listed on this screen, and that they correspond with those shown in the prompter design work area.

For the state codes prompter, you must change both the tab character (from @ to /) and settings (shown below) by entering the new tab character and settings over the old ones as indicated:

```
PRD006                          M  A  N  T  I  S

                            Prompter Design Facility



                  Tab character  ..................... /

                  Tab positions  ..................... 29
                                                      57
```

Use the space bar or Erase EOF key to erase the extra five entries.

When you have made the changes indicated above, press ENTER. MANTIS automatically returns you to the Prompter Design Facility menu.

Step 2: Creating the prompter

Now that the tabs are specified correctly, you are ready to select the Create or update a prompter option. This option lets you create a new prompter design or update an existing prompter design.

To select the Create or update a prompter option from the Prompter Design Facility menu, enter 1 in the action field and press ENTER. MANTIS displays the prompter design work area:

```
PRD002  Page = 1              Create Or Update A Prompter        Lines =
   ....|....1....|....2....|.../3....|....4....|....5....|./..6....|....7....|..
```

Notice that the scale line across the top of the prompter design work area now reflects the new tab character (/) and tab settings (29 and 57).To manipulate information in the prompter design work area, you use the A (Alter), I (Insert), and D (Delete) action indicators, as you did during file design.

Enter these indicators in the first tab position of the row you want to alter, insert, or delete. You then enter the actual text of the prompter beginning in the second tab position.

You are now ready to begin entering the prompter data. Use the Tab key to move the cursor to the beginning of the first line, then enter the data as shown:

```
PRD002  Page = 1              Create Or Update A Prompter        Lines =0
   ....|....1....|....2....|.../3....|....4....|....5....|./..6....|....7....|..
i al - alabama/ky - kentucky/nd - north dakota
i ak - alaska/la - louisiana/oh - ohio
i az - arizona/me - maine/ok - oklahoma
i ar - arkansas/md - maryland/or - oregon
i ca - california/ma - massachutsetts/pa - pennsylvania
i co - colorado/mi - michigan/ri - rhode island
i ct - connecticut/mn - minnesota/sc - south carolina
i de - delaware/ms - mississippi/sd - south dakota
i dc - district of columbia/mo - missouri/tn - tennessee
i fl - florida/mt - montana/tx - texas
i ga - georgia/ne - nebraska/ut - utah
i hi - hawaii/mv - nevada/vt - vermont
i id - idaho/nh - new hampshire/va - virginia
i il - illinois/nj - new jersey/wa - washington
i in - indiana/nm - new mexico/wv - west virginia
i ia - iowa/ny - new york/wi - wisconsin
i ks - kansas/nc - north carolina/wy - wyoming
```

As you enter the state data, experiment with the A (ALTER) and D (DELETE) commands, as well as with I (INSERT).

After you have entered all of the data, press ENTER. MANTIS evaluates your entries and displays the prompter according to the tabs you specified:

```
PRD002  Page = 1              Create Or Update A Prompter        Lines =17
   ....|....1....|....2....|.../3....|....4....|....5....|./..6....|....7....|..
AL - ALABAMA                 KY - KENTUCKY              ND - NORTH DAKOTA
AK - ALASKA                  LA - LOUISIANA             OH - OHIO
AZ - ARIZONA                 ME - MAINE                 OK - OKLAHOMA
AR - ARKANSAS                MD - MARYLAND              OR - OREGON
CA - CALIFORNIA              MA - MASSACHUTSETTS        PA - PENNSYLVANIA
CO - COLORADO                MI - MICHIGAN              RI - RHODE ISLAND
CT - CONNECTICUT             MN - MINNESOTA             SC - SOUTH CAROLINA
DE - DELAWARE                MS - MISSISSIPPI           SD - SOUTH DAKOTA
DC - DISTRICT OF COLUMBIA    MO - MISSOURI              TN - TENNESSEE
FL - FLORIDA                 MT - MONTANA               TX - TEXAS
GA - GEORGIA                 NE - NEBRASKA              UT - UTAH
HI - HAWAII                  NV - NEVADA                VT - VERMONT
ID - IDAHO                   NH - NEW HAMPSHIRE         VA - VIRGINIA
IL - ILLINOIS                NJ - NEW JERSEY            WA - WASHINGTON
IN - INDIANA                 NM - NEW MEXICO            WV - WEST VIRGINIA
IA - IOWA                    NY - NEW YORK              WI - WISCONSIN
KS - KANSAS                  NC - NORTH CAROLINA        WY - WYOMING
```

Notice that wherever you designated a tab character, MANTIS moves the data directly following that tab character to the next tab position.

MANTIS also supplies the current page number (you can have up to four pages in a prompter), and the number of lines displayed on the current page. If your prompter has entries filling more than one page, you can page through them by entering the page number after PAGE, or by using the PF1–PF4 keys.

Press CANCEL to return to the Prompter Design Facility menu.

Step 3: Saving the prompter

You must now save the prompter design. Select the Library functions option from the Prompter Design Facility menu by entering 3 in the action field and pressing ENTER. MANTIS displays the Prompter Design Library Facility menu:

```
PRD004                          M  A  N  T  I  S


                      Prompter Design Library Facility



   Name  ............... :                                                  :
   Description  ........ :                                                 :
   Password  ........... :                  :
   Next Prompter  ...... :                  :
   Language  ........... : ENGLISH          :
   Export File Name .... :                                    :

                       Save  ...................  1
                       Replace  .................  2
                       Fetch  ...................  3
                       Delete  ..................  4
                       Export  ..................  5
                       Import  ..................  6
                       Exit  ................  CANCEL

                                 :   :
```

If you have an existing prompter design in your work area, the name and description of that prompter displays. (The password does not display.)

Since this is a new prompter, you must supply a name, description, and password to save the design. Enter the information as shown (continue to use your user ID as the password, although it will not display as you enter it):

```
PRD005                          M  A  N  T  I  S


                        Prompter Design Library Facility



   Name  ............... : STATE_CODES                                    :
   Description  ........ : State codes                                  :
   Password  ........... : PASSWORD          :
   Next Prompter  ...... :                   :
   Language  ........... : ENGLISH           :
   Export File Name .... :                                  :

                       Save  ...................  1
                       Replace  .................  2
                       Fetch  ...................  3
                       Delete  ..................  4
                       Export  ..................  5
                       Import  ..................  6
                       Exit  ................  CANCEL

                                 : 1 :
```

You can use the Chain to next prompter field to make the prompter longer than four pages. (The additional prompter can be named here and created later.) However, for the state codes prompter, leave this field blank.

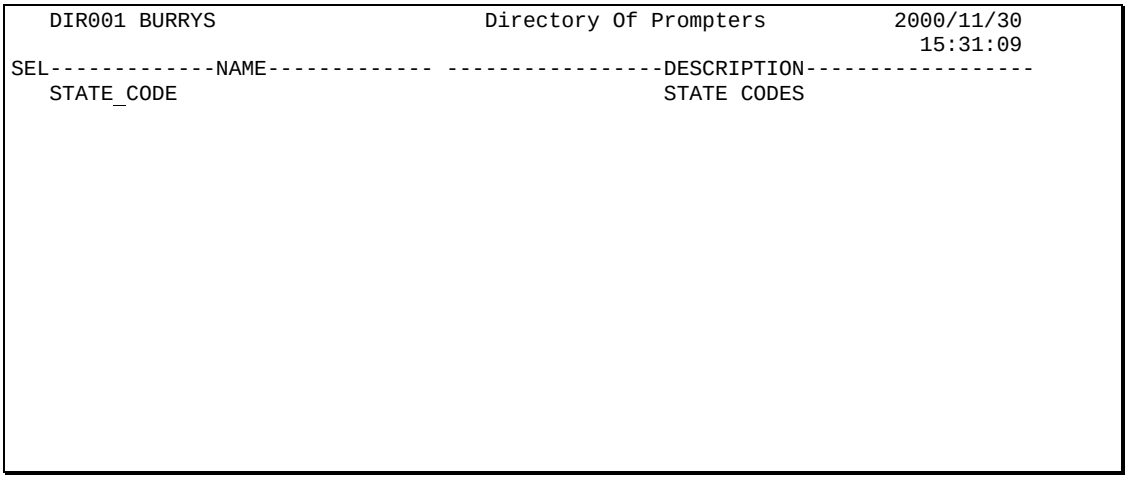
The description you provide will become the centered title for the prompter when the final design is displayed. It will also appear in your directory.

Save the prompter by entering 1 in the action field and pressing ENTER, or by pressing PF1. MANTIS automatically returns you to the Prompter Design Facility menu and displays a confirmation message in the lower, left corner of the screen.

Step 4: Viewing the directory of prompters

Use the Directory of prompters option to display an alphabetic listing of all existing prompters.

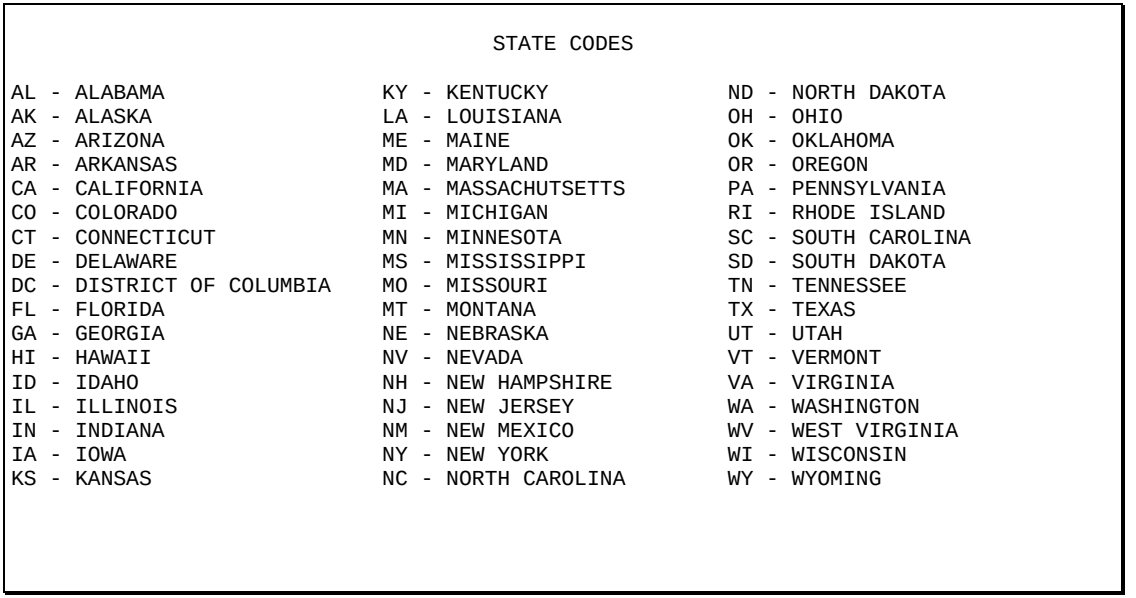
To display the list, select the Directory of prompters option from the Prompter Design Facility menu by entering 4 in the action field and pressing ENTER (or by pressing PF4). MANTIS displays the Directory of Prompters screen:



Press ENTER to return to the Prompter Design Facility menu.

Step 5: Displaying the completed prompter design

You can check the final prompter design by selecting the Display completed design option. On the Prompter Design Facility menu, enter 5 in the action field and press ENTER. MANTIS displays the prompter currently in your work area:



```
                                STATE CODES

AL - ALABAMA                 KY - KENTUCKY              ND - NORTH DAKOTA
AK - ALASKA                  LA - LOUISIANA             OH - OHIO
AZ - ARIZONA                 ME - MAINE                 OK - OKLAHOMA
AR - ARKANSAS                MD - MARYLAND              OR - OREGON
CA - CALIFORNIA              MA - MASSACHUTSETTS        PA - PENNSYLVANIA
CO - COLORADO                MI - MICHIGAN              RI - RHODE ISLAND
CT - CONNECTICUT             MN - MINNESOTA             SC - SOUTH CAROLINA
DE - DELAWARE                MS - MISSISSIPPI           SD - SOUTH DAKOTA
DC - DISTRICT OF COLUMBIA    MO - MISSOURI              TN - TENNESSEE
FL - FLORIDA                 MT - MONTANA               TX - TEXAS
GA - GEORGIA                 NE - NEBRASKA              UT - UTAH
HI - HAWAII                  NV - NEVADA                VT - VERMONT
ID - IDAHO                   NH - NEW HAMPSHIRE         VA - VIRGINIA
IL - ILLINOIS                NJ - NEW JERSEY            WA - WASHINGTON
IN - INDIANA                 NM - NEW MEXICO            WV - WEST VIRGINIA
IA - IOWA                    NY - NEW YORK              WI - WISCONSIN
KS - KANSAS                  NC - NORTH CAROLINA        WY - WYOMING
```

Your completed prompter design should correspond with the one above. Notice that the description has become the title for the prompter.

You can return to the Prompter Design Facility menu at any time during the prompter design phase and select the Display completed design option to view the prompter design.

Press ENTER to move through multiple pages of the prompter display. When you are finished, press ENTER or CANCEL to end the prompter display and return to the Prompter Design Facility menu.

Step 6: Printing the completed prompter design

If you would like to view a printed copy of the prompter design, select the Print completed design option. (Enter 6 in the action field and press ENTER.) The printout will be routed to your designated printer.

Exercise

Since the state codes prompter you just created is the only prompter for the Burrys scenario, there are no exercises for this chapter.

You are now ready to proceed to chapter 5, where you will learn MANTIS programming fundamentals.

5

Understanding MANTIS programming fundamentals

Before you begin working on the Burrys menu program, you'll need to learn some basic features of the MANTIS programming language and the MANTIS Program Design Facility. This chapter introduces you to these concepts, then guides you step-by-step through the process of creating the menu program for the Burrys application.

Learning outline

In this chapter you will learn how to:

- Understand basic conventions of the MANTIS language, including the number set, character set, symbolic names, numeric and text expressions, and program comments
- Access the Program Design Facility menu and select options on it
- Use the EDIT Program Entry screen to specify a name and description for your program
- Edit your program

Basic concepts: Understanding MANTIS language conventions

Many conventions of the MANTIS language are discussed throughout the programming lessons in this tutorial. However, before you begin to write a program, you should review the basic conventions presented in the following sections.

MANTIS number set

The MANTIS number set consists of:

- Digits 0–9
- Preceding plus or minus sign
- Period (decimal point)
- Letter E

Internally, MANTIS stores numeric data in floating point (scientific notation) and regards numbers in one of two ways:

- **SMALL**. Stores numbers up to 6 significant digits
- **BIG**. Stores numbers up to 14 significant digits

Even if your installation uses a decimal point character other than the period (.) in user screens (for example, the comma), you must use a period for a decimal point in the numbers in your programs.

MANTIS character set

The MANTIS character set consists of:

- Alphabetic characters A–Z
- Space character
- Numeric digits 0–9
- Special characters, such as the underline (_) and vertical bar (|).

Understanding literals, symbolic names, variables, and expressions

Before you begin creating MANTIS programs, you should understand the following terms:

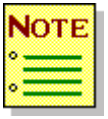
- A *literal* is a direct representation of a value (for example, 4 or “ACCOUNT”). In MANTIS, a numeric literal can consist of the numbers 0 through 9, the plus sign (+), and the minus sign (-). A text literal can consist of a quoted string of any valid ASCII characters.
- A *symbolic name* is a string of characters that represents a user-defined object in a MANTIS program (for example, MAP, CREDIT_LIMIT). MANTIS uses symbolic names to represent variables processed by MANTIS programs. A symbolic name can represent a complex entity such as a screen or file, or it can represent a field.
- A *variable* is a symbolic name that contains a value. Variables can have different values at different points in time (for example X, where X can be any number). Variables can represent a single value (a scalar), a text array, or a numeric array of values.
- An *expression* is a combination of text or numeric variables, functions, and/or literals that can be evaluated by MANTIS.

Using symbolic names

Symbolic names can represent either numeric or text data. MANTIS allows a maximum of 65536 symbolic names for a single program, including names defined indirectly by SCREEN, FILE, and ACCESS statements.

A symbolic name:

- Must begin with an alphabetic character.
- Can contain alphabetic characters, numeric characters, and the underline (_). *No spaces or other special characters are allowed.*
- Can be entered as lowercase letters; MANTIS will convert them to uppercase (for example, the following symbolic names are equivalent: customer_name, Customer_Name, and CUSTOMER_NAME).
- Can be any size that fits on a line. However, if the field is used in a design entity (that is, screens, interfaces, or files), the symbolic name is limited to 16 , 30 or 80 characters, depending upon the entity.
- Must be unique. If MANTIS encounters a definition for a symbolic name for the second time (a duplicate definition), it ignores the second definition and does not create a second work area for that name.
- Can contain a reserved word (for example, EDITOR), but cannot be a reserved word in its entirety (for example, EDIT).



Do not use hyphens to connect words in symbolic names. If you do, MANTIS will try to subtract the value of the second word from the first.

The following table shows examples of symbolic names, indicating whether each is valid or invalid:

Symbolic name	Valid/invalid
INVESTMENT	Valid
COST_OF_SALES	Valid
SALES_2000	Valid
2000_SALES	Invalid; must begin with alphabetic character
PROGRAM	Invalid; PROGRAM is a reserved word
MENU_PROGRAM	Valid
CUST NAME	Invalid; must not contain blank spaces
CREDIT-LIMIT	Invalid; must not contain a hyphen

Using literals, variables, and expressions

MANTIS supports numeric and text literals, variables, and expressions as described in the following sections.

Numeric literals, variables, and expressions

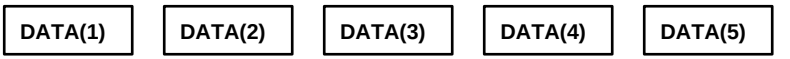
Numeric literals consist of the digits 0 through 9 and can have the unary signs plus and minus (for example, -9). Numbers may optionally contain one decimal point (for example, 3.14). Numbers may also use the e-notation (exponential). For example, 6.02257E+23 means 6.02257 times 10 to the 23rd power.

Numeric variables consist of a valid MANTIS symbolic name. A numeric variable can represent a single entity (for example SUM, where SUM contains a number) or a numeric array.

MANTIS can store numeric values in ordered sets called *arrays*. Arrays can have one or two dimensions. You can specify arrays as BIG or SMALL. For example, if you specify a one-dimensional array as follows:

```
BIG DATA(5)
```

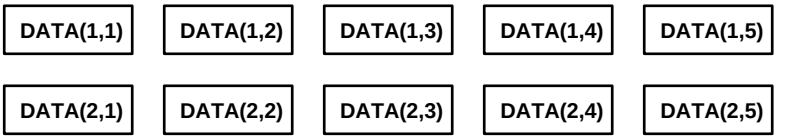
MANTIS allocates storage for five BIG occurrences of DATA:



If you specify a two-dimensional array of two rows and five columns:

```
BIG DATA(2,5)
```

MANTIS allocates storage as a two-dimensional array:



If the variable represents an array, individual elements can be referenced by subscripting the variable name, for example MONTHLY_SALES(3) represents the value for the third element of the array.

Numeric expressions consist of a combination of numeric variables and/or literals (for example, FINAL_SUM=SUM+9).

Text literals, variables, and expressions

A *text literal* is any set of 0-254 ASCII characters enclosed in quotes (for example, “THIS IS A TEXT LITERAL”). Use two consecutive quotes to specify one occurrence of a quotation mark within a text literal. For example, if you issue the following command:

```
SHOW "THE ANSWER IS ""Y"""
```

MANTIS returns the following:

```
THE ANSWER IS "Y"
```

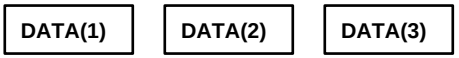
A *text variable* is a symbolic name that can assume up to 254 ASCII characters (letters, numerals, punctuation, native language characters, etc.). The default for a variable is numeric (see BIG), so you *must* define text variables before using them. If you specify:

```
TEXT DATA(5)
```

MANTIS allocates 5 characters of memory for the variable, DATA. You can also define a text array in your program. For example, if you enter the statement:

```
TEXT DATA(3,10)
```

MANTIS allocates storage as:



where each DATA element has 10 bytes (characters) of storage available.

Your installation may have a different size and subscript limit on TEXT variables, if changed by the MASTER USER.

Text expressions consist of a combination of text variables and/or literals (for example, ACCOUNT_INFO="ACCOUNT "+ACCOUNT_NUMBER).

Defining variables in a program

You can define a variable in a program in the following ways:

- With variable definitions within a SCREEN, FILE, INTERFACE, TOTAL, ACCESS, or VIEW layout. The variable becomes defined when the statement associated with the complex entity is executed. (If text, the variable is initially a zero-length string; if numeric, the variable is initially zero.)
- With a TEXT, BIG, SMALL, or KANJI statement. (If text, the variable is initially a zero-length string; if numeric, the variable is initially zero.) If a symbolic name is not yet defined when it is first used, it is assigned a default type of BIG, with initial value of zero.
- As a parameter on the program's main entry statement where the argument was passed. (The variable has the definition and the actual data of the invoking program's parameter.)

Step-by-step: Creating the customer menu program

This section provides a step-by-step introduction to MANTIS program design, including:

- Accessing the Program Design Facility
- Using the EDIT Program Entry screen
- Creating a program in the external editor

MANTIS programs are created and managed in *programming mode*. To enter MANTIS programming mode, select the Design a Program option from the MANTIS Facility Selection menu. The Program Design Facility menu displays:

```
PRG001                          M  A  N  T  I  S

                           PROGRAM DESIGN FACILITY

                    Update Program  .............(S).......  1
                    Directory Of Programs  ................  2
                    Library Functions  ....................  3
                    Print Program  ..............(L).......  4
                    Bind Program  ...............(B).......  5
                    Unbind Program  .............(U).......  6
                    Purge Program  ..............(P).......  7
                    Purge Program Source  .......(N).......  8
                    Edit Program  ...............(E).......  9
                    Exit Program Design  .............  CANCEL

  Load Program .. :                                                 :
  Current Program : EXAMPLES:EXAMPLES
```

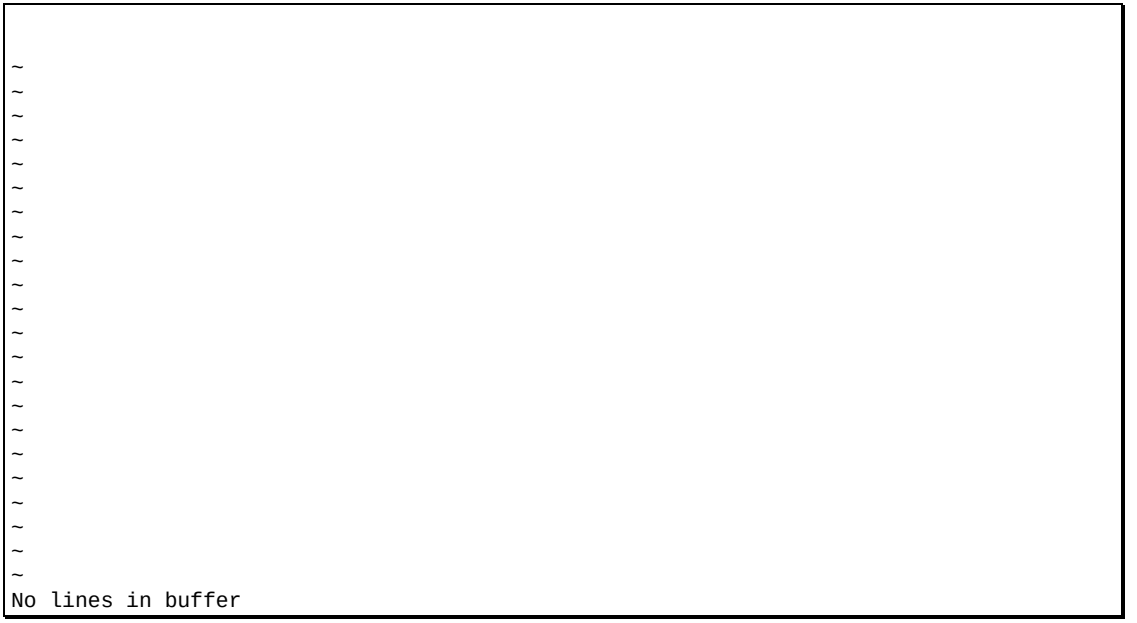
From the Program Design Facility menu, you can select options to create, maintain, view, print, bind, unbind, and edit MANTIS programs. To create the Burrys menu program, you will use options from the Program group to access the external editor and perform other tasks related to creating and maintaining MANTIS programs.

Step 1: Using the EDIT Program Entry screen

From the Program Design Facility menu, enter 9 to select Edit Program.

```
PRG001                          M  A  N  T  I  S

                              PROGRAM DESIGN FACILITY

                     Update Program  .............(S).......  1
                     Directory Of Programs  ................  2
                     Library Functions  ....................  3
                     Print Program  ..............(L).......  4
                     Bind Program  ...............(B).......  5
                     Unbind Program  .............(U).......  6
                     Purge Program  ..............(P).......  7
                     Purge Program Source  .......(N).......  8
                     Edit Program  ...............(E).......  9
                     Exit Program Design  ..............  CANCEL

  Load Program .. :                                                 :
  Current Program : EXAMPLES:EXAMPLES

                                       : 9  :
```

Press ENTER. In OpenVMS, EDT is the default external editor. In UNIX, vi is the default external editor. It is assumed that the reader is already familiar with EDT or vi. The vi editor screen displays:



Once you enter the external editor, you can create and modify MANTIS programs.

Step 2: Entering a program in the external editor

To create the new menu program:

1. Enter the following program statements. *Do not enter line numbers*:

```
ENTRY CUST_MENU
SCREEN MAP("CUST_MENU")
CONVERSE MAP
WHILE MAP<>"CANCEL"
WHEN ACTION=1 OR MAP="PF1"
SHOW"PROGRAM CHAINS TO CUSTOMER ENTRY SCREEN"
WAIT
WHEN ACTION=2 OR MAP="PF2"
SHOW"PROGRAM CHAINS TO CUSTOMER BROWSE SCREEN":WAIT
END
CLEAR MAP
CONVERSE MAP
END
EXIT
```

When you've finished entering the program lines, your screen should look like this:

```
ENTRY CUST_MENU
SCREEN MAP("CUST_MENU")
CONVERSE MAP
WHILE MAP<>"CANCEL"
WHEN ACTION=1 OR MAP="PF1"
SHOW"PROGRAM CHAINS TO CUSTOMER ENTRY SCREEN"
WAIT
WHEN ACTION=2 OR MAP="PF2"
SHOW"PROGRAM CHAINS TO CUSTOMER BROWSE SCREEN":WAIT
END
CLEAR MAP
CONVERSE MAP
END
EXIT

~
~
~
~
~
~
~
~
No lines in buffer
```

2. Use OpenVMS EDT or UNIX vi command to exit from the external editor. MANTIS copies the edited program back into the work area. MANTIS will automatically sequence the program when you exit from the editor. The automatic sequencing line increment value is one.

3. Enter the following command; then, press ENTER:

   ```
   SEQUENCE 100,10
   ```

   MANTIS renumbers your program line numbers, beginning at line 100 and incrementing the line numbers by ten.

4. Enter SAVE CUST_MENU and press ENTER. The SAVE command saves the new program in your library.

You have just created and saved your first MANTIS program! In the next chapter, you'll take a closer look at this program, examining the code step-by-step to learn the purpose of each line. For now, just get a general sense of how a MANTIS program is constructed.

Exercises

Complete the exercises in this section before you proceed to the next chapter.

Exercise 1: Verifying the menu program

Examine the menu program that you just created and make sure that it's identical to this one:

Use the LIST command to list program lines in the Program Design work area.

```
100 ENTRY CUST_MENU
110 .SCREEN MAP("CUST_MENU")
120 .CONVERSE MAP
130 .WHILE MAP<>"CANCEL"
140 ..WHEN ACTION=1 OR MAP="PF1"
150 ...SHOW"PROGRAM CHAINS TO CUSTOMER ENTRY SCREEN"
160 ...WAIT
170 ..WHEN ACTION=2 OR MAP="PF2"
180 ...SHOW"PROGRAM CHAINS TO CUSTOMER BROWSE SCREEN":WAIT
190 ..END
200 ..CLEAR MAP
210 ..CONVERSE MAP
220 .END
230 EXIT
```

If your program is not identical to the one above, update your program by using the external editor.

If you need to add a program line, enter the line number and program statement on the command line; then, press ENTER to add the line to your program. To delete an extra line from your program, enter the line number you want to delete; then, press ENTER.

When you are finished, save your changes by entering REPLACE on the command line and pressing ENTER.

Exercise 2: Adding records to the state codes file

Now, to apply what we've discussed so far in this tutorial, let's create another simple program, run it, and enter records into a file.

To begin, you will need to clear the program work area so that you can enter a new program. To do so, enter NEW on the command line and press ENTER. Follow the procedures discussed in this chapter to create and save the following program:

```
100 ENTRY STATE_CODES
110 .SCREEN MAP("STATE_CODE")
120 .FILE REC("STATE_CODES","PASSWORD")
130 .CONVERSE MAP
140 .WHILE MAP<>"CANCEL"
150 ..INSERT REC
160 ..CLEAR MAP
170 ..CONVERSE MAP
180 .END
190 EXIT
```

To save your program, enter SAVE on the program work area command line; then, press ENTER.

Now, enter RUN on the command line and press ENTER. MANTIS displays the state code entry screen (STATE_CODE) that you created in chapter 2.

In the CUST_STATE field, enter a state code abbreviation and press ENTER. MANTIS adds the state code to the STATE_CODES file. Add four or five more state codes to the file. (We'll use them in a later exercise.) When you're finished, press CANCEL to terminate the program and return to the program work area.

Exercise 3: Adding records to the customer information file

Now, let's add a few records to the customer information file, so that we can use them in a future exercise. To do so, we'll create another program, run it, and enter records into the file.

Use the procedures discussed in this chapter to create and save the following program:

```
100 ENTRY ADD_CUST
110 .SCREEN MAP("CUST_ENTRY")
120 .FILE REC("CUST_INFO","PASSWORD")
130 .CONVERSE MAP
140 .WHILE MAP<>"CANCEL"
150 ..INSERT REC
160 ..CLEAR MAP
170 ..CONVERSE MAP
180 .END
190 EXIT
```

Next, enter RUN on the command line and press ENTER. MANTIS displays the customer entry screen (CUST_ENTRY) that you created in chapter 2.

Enter a customer record; then, press ENTER. MANTIS adds the customer record to the CUST_ENTRY file. (Because MANTIS automatically maps data between like-named fields on the screen and file record, no program code is required to move the data.) Add several more customer records to the file; then, press CANCEL to terminate the program and return to the program work area.

Enter QUIT to return to Program Design Facility menu.

These exercises demonstrate how easily and quickly you can create applications using MANTIS. In the following chapters, you'll learn more about MANTIS programming statements and commands, and how to use them to build more complex programs.

6

Using MANTIS programming statements and commands

Now that you know how to create a program, it's time to learn more about the MANTIS programming environment. This chapter provides an overview of all the programming-mode commands, and shows you how to use them to modify the menu program that you created in the previous chapter.

Learning outline

In this chapter you will learn how to:

- Understand the difference between programming statements and immediate mode statements
- Understand how to use basic MANTIS programming statements
- Understand MANTIS commands and use them to edit a program in Program Design (programming mode).
- Create program stubs for the CUST_ENTRY and CUST_BROWSE programs.

Basic concepts: Understanding statements and commands

The MANTIS programming environment is comprised of these elements:

- **Statements.** There are two types of MANTIS statements:

  - **Program statements.** A *program statement* consists of a line number, a MANTIS language element, and possibly one or more operands. As part of a program, program statements require a run-mode action; that is, they are not executed until the program is run.

    MANTIS automatically indents program statements to indicate their relative position in the program's nesting hierarchy. The following line shows an example of a program statement:

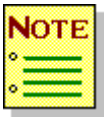
    ```
    300 FILE REC("MYFILE","MYFILEPSWD")
    ```

  - **Immediate mode statements.** An *immediate mode statement* is one that is entered without a line number, on the command line, indicating that it should be interpreted and executed immediately and not become a part of the program.

    While an immediate mode statement uses statements that are also valid in a MANTIS program, not all program statements can be entered in immediate mode. (For example, logic statements such as IF, WHEN, and WHILE are not valid.) So, if entered on the command line, the following statement will be executed immediately:

    ```
    FILE REC("MYFILE","MYFILEPSWD")
    ```

- **Commands.** A *command* is not part of a program, but is an order that you give to MANTIS when you want it to perform some action (such as SAVE, QUIT, EDIT, etc.). MANTIS uses editing commands to maintain programs. A command designates an immediate mode action, and is executed immediately. A command cannot have a line number.



Some MANTIS reserved words can be used as both statements and commands (such as DISPLAY, SHOW).

Using MANTIS program statements

To gain a better understanding of MANTIS program statements, let's take a closer look at the Burrys menu program that you created in chapter 5:

Use the LIST command to list program lines in the Program Design work area.

```
100 ENTRY CUST_MENU
110 .SCREEN MAP("CUST_MENU")
120 .CONVERSE MAP
130 .WHILE MAP<>"CANCEL"
140 ..WHEN ACTION=1 OR MAP="PF1"
150 ...SHOW"PROGRAM CHAINS TO CUSTOMER ENTRY SCREEN"
160 ...WAIT
170 ..WHEN ACTION=2 OR MAP="PF2"
180 ...SHOW"PROGRAM CHAINS TO CUSTOMER BROWSE SCREEN":WAIT
190 ..END
200 ..CLEAR MAP
210 ..CONVERSE MAP
220 .END
230 EXIT
```

ENTRY-EXIT statements

The ENTRY and EXIT statements define the boundaries of your program:

```
100 ENTRY CUST_MENU
230 EXIT
```

Use the ENTRY-EXIT statements around programs. A symbolic name (in this case, CUST_MENU) must follow ENTRY. Remember that a MANTIS symbolic name must begin with a letter and must use the underscore to connect parts of the name. (Do not use blanks to connect parts of the name, or MANTIS will treat the parts as separate symbolic variables.)

SCREEN statement

In line 110, the menu program specifies the screen to be used:

```
110 .SCREEN MAP("CUST_MENU")
```

As part of the SCREEN statement, you must supply a symbolic name (in this case, MAP) as well as the name you gave your screen when you saved it (“CUST_MENU”). Enclose the screen name in quotes and parentheses.

Some MANTIS sites implement standard naming conventions that require the entity name to be prefixed with the library name (that is, the user ID name) in all program statements that specify MANTIS entities (SCREEN, FILE, etc.). This practice allows the program to be copied to, and run from, another MANTIS user while the other entities reside on the original user.

To include the library name, use the syntax “library:entity_name”. For example, using this syntax the preceding SCREEN statement would read: `110 .SCREEN MAP(“BURRYS:CUST_MENU”).`

Although the SCREEN statement identifies a screen to your program and assigns it a symbolic name, the SCREEN statement does not actually display the screen.

CONVERSE statement

In line 120, the program displays the screen that you specified in the SCREEN statement:

```
120 .CONVERSE MAP
```

The CONVERSE statement tells the program to display a particular screen. Use the CONVERSE statement with the screen’s *symbolic* name. The CONVERSE statement also moves any user input from the terminal into the defined screen variables.

When you execute the menu program, CONVERSE sends the screen to the terminal. You can then select one of the options on the screen (1, 2, or CANCEL), and MANTIS returns your response to the program, either in your screen variable (ACTION) or the symbolic name of the screen (MAP) if a PF key is pressed.

WHILE-END and WHEN-END

Two basic structures control the flow of the menu program: a WHILE loop and a WHEN conditional. Loops and conditionals form the building blocks for MANTIS programs.

The WHILE loop, defined by WHILE-END statements, executes a block of statements repeatedly as long as the specified condition is true:

```
130 .WHILE MAP<>"CANCEL"
...
220 .END
```

The symbol <> is a MANTIS operator that means “does not equal.” Thus, your program will execute any statements that you place between lines 130 and 220, until the user presses CANCEL (the key value placed in the MAP symbolic name variable).

In the menu program, the WHILE test essentially asks the question, “You haven’t pressed CANCEL, have you?” As long as MANTIS receives the answer “no,” it executes statement 140:

```
140 ..WHEN ACTION=1 OR MAP="PF1"
...
170 ..WHEN ACTION=2 OR MAP="PF2"
...
190 ..END
```

The WHEN conditional statements in the menu program provide for the two remaining options on your menu. Remember that the data field on the menu screen is named ACTION, and that it’s a numeric field. You want the user to be able to select an option by keying in 1 or 2 and pressing ENTER, or pressing the corresponding PF key.

WHEN-END executes a block of statements only when a condition you specify is true. In the menu program, MANTIS asks at statement 140 whether or not you entered 1 or pressed PF1. If so, MANTIS executes lines 150 and 160 and then performs the next WHEN test at 170; otherwise, it performs the next WHEN test at 170. You can include any number of WHEN tests in a series. Note also that the WHEN conditional structure is not terminated when the first WHEN condition is met; all of them are checked.

SHOW and WAIT statements

The program contains SHOW and WAIT statements that we used as a place-holder for the actual statements that we want to insert there.

The SHOW statements (lines 150 and 180) display data that you specify on an unformatted screen. For example, the SHOW statement in line 150 displays the text string “PROGRAM CHAINS TO CUSTOMER ENTRY SCREEN”:

```
150 ...SHOW"PROGRAM CHAINS TO CUSTOMER ENTRY SCREEN"
160 ...WAIT
...
180 ...SHOW"PROGRAM CHAINS TO CUSTOMER BROWSE SCREEN":WAIT
```

The colon (:) on line 180 tells MANTIS that another statement (in this case, WAIT) appears directly following it. So, line 180 performs the similar action as lines 150 and 160 combined.

A WAIT statement temporarily suspends execution of your program by sending any pending SHOW data to the screen and waiting for an operator response. In this example, if you enter the RUN command on the command line and press ENTER, and then when the CUST_MENU screen displays press PF1 (or enter 1 and press ENTER), your program will display the first text string and pause:

```
PROGRAM CHAINS TO CUSTOMER ENTRY SCREEN
```

If you had pressed PF2 instead, the other text string would have displayed:

```
PROGRAM CHAINS TO CUSTOMER BROWSE SCREEN
```

Press ENTER to return to the CUST_MENU screen; then, press CANCEL to return to the program work area.

You can use SHOW and WAIT statements as a debugging tool, by placing them at strategic locations throughout a MANTIS program.

CLEAR statement

After an entry other than CANCEL, the menu program executes the CLEAR statement to erase the data in the ACTION field (line 200). The CLEAR statement also erases the value in the symbolic name MAP, if a PF key was pressed:

```
200 ..CLEAR MAP
210 ..CONVERSE MAP
```

Supply the screen's symbolic name (in this case, MAP) after CLEAR, to tell MANTIS which screen to clear.

The menu program follows the CLEAR statement with another CONVERSE statement to display the screen and await the next entry in the WHILE loop.

Including comments in a program

You can include comments in a program by entering a vertical bar followed by descriptive text:

```
|THIS IS A SAMPLE COMMENT.
```

MANTIS ignores comments during program execution, so do not place program statements after the vertical bar. The one exception to this rule is that MANTIS does recognize SQL database access statements after the comment bar, when they are enclosed within the EXEC_SQL and END statements. (For more detailed information on coding the EXEC_SQL-END structure, refer to AD/Advantage MANTIS Language OpenVMS/UNIX, P39-1310.)

Using MANTIS programming commands

A MANTIS programming command is a reserved word *without* a line number. MANTIS does not consider a command to be part of your program and executes it immediately. For example, you use commands to SEQUENCE or RUN a program. (For a detailed explanation of the syntax of MANTIS programming commands, refer to AD/Advantage MANTIS Facilities OpenVMS/UNIX, P39-1300.)

Programming commands can be program design commands or program statements that are executed as commands.

Using program design commands

When you enter MANTIS commands at the bottom of the screen without a line number, MANTIS will execute them immediately. This table provides a quick reference for the MANTIS program design commands.

Command	Description
ALTER	Present specified program line(s) for modification
BIND	Converts a program from unbound to bound format (BIND [ON]), or from bound to unbound format (BIND OFF)
CHANGE	Replace occurrences of a string of characters in a range of program lines
CLEAR BREAK	Clear program breakpoints
COPY	Specifies that MANTIS should copy all or part of a MANTIS program into the program being edited
DISPLAY	Displays attributes of MANTIS variables or MANTIS options
DOWN	Move down to a lower-level routine
EDIT	Invoke an external text editor
ERASE	Deletes one or more program lines
GO	Resume execution from a program breakpoint
HELP	Displays a help prompter for an error code, a command, a reserved word, or online help
KILL	Terminates a program listing, a paused program or a program in a loop
LIST	Lists all or part of the program currently in work area

Command	Description
LOAD	Retrieves a program from a library
NEW	Clears the current work area so that you can enter a new program
PURGE	Erases the program from your library
QUIT	Terminates programming mode and return to the Program Design Facility Menu
RUN	Executes the program currently in the work area
SAVE	Copies the edited program into the library
SEQUENCE	Renumbers the lines of the program that you are editing
USAGE	Shows where a MANTIS keyword, symbolic name, or string is used in your program
UP	Move up to a higher-level routine

For more information about the format of each command and guidelines for usage, refer to AD/Advantage MANTIS Language OpenVMS/UNIX, P39-1310.

Using statements as commands

You can execute some statements as commands by entering them without a statement number. Such a statement is not part of the coded program and MANTIS executes it immediately. In the following example, SHOW appears as both a statement and a command. First, in this program SHOW appears as a *statement*:

```
10  ENTRY COMPOUND
20  .SHOW"THIS PROGRAM DEMONSTRATES THE SHOW STATEMENT."  <-statement
30  .WAIT
40  .LET INVESTMENT=1400
50  EXIT
```

If you enter the RUN command and press ENTER, MANTIS displays the following:

```
THIS PROGRAM DEMONSTRATES THE SHOW STATEMENT.
```

If you then press ENTER to return to program design, you can enter the SHOW statement as a *command* on the bottom of the screen and press ENTER:

```
SHOW INVESTMENT+650              <-command
```

The above command tells MANTIS to add 650 to INVESTMENT (which was assigned a value of 1400 in line 40 of the program), and displays the following result above the command line:

```
2050
```

You can see how using statements as commands might help you to test and debug a program. You can stop the program at various points and view or assign variables to help determine how your program is executing.

Step-by-step: Completing the menu program

To complete the Burrys menu program, we'll add a comment and replace the SHOW and WAIT statements with CHAIN statements.

Since the MANTIS development environment is interactive and programs can be executed immediately without compiling, it's easy to develop a program in a series of steps, testing each step along the way. This process is called *step-wise refinement*, because each step adds more detail and function to the program.

The interactive nature of the MANTIS development environment also makes it easy to demonstrate partially completed programs to end-users to ensure that the finished application will meet their needs. This process is called *prototyping*. The programs you will be developing in the following sections will apply the step-wise refinement and prototyping techniques.

To begin, select Design a Program from the Facility Selection menu. At the Load Program field on the Program Design Facility menu, enter CUST_MENU ; then, press ENTER:

```
PRG001                          M  A  N  T  I  S

                              PROGRAM DESIGN FACILITY

                     Update Program  .............(S).......  1
                     Directory Of Programs  ................  2
                     Library Functions  ....................  3
                     Print Program  ..............(L).......  4
                     Bind Program  ...............(B).......  5
                     Unbind Program  .............(U).......  6
                     Purge Program  ..............(P).......  7
                     Purge Program Source  .......(N).......  8
                     Edit Program  ...............(E).......  9
                     Exit Program Design  .............  CANCEL

  Load Program .. :                                                 :
  Current Program : BURRYS:CUST_MENU

                                      :    :
```

'BURRYS:CUST_MENU' FetchedMANTIS loads your program into Program Work Area. Press GOLD/1 keys to start editing the program.

Step 1: Adding a comment

First, we'll add a comment line after 110.

```
115 | This is the customer accounts menu
```

Remember, comments begin with a vertical bar and include the rest of the line. When you press ENTER, MANTIS inserts line 115.

You can also include comments on the same line with a program statement by entering a colon (:), a vertical bar (|), and your comment after the statement:

```
10  ENTRY CUST_MENU:|THIS IS AN EXAMPLE
```

Because MANTIS ignores comments when executing a program (except for SQL statement text), don't place statements or commands on the same line after a comment.

Step 2: Finding and changing a program line

Next, we want to change the SHOW statements to CHAIN statements. We will use the USAGE command to find the SHOW statements, so that we can change them.

The USAGE command finds text in your program. MANTIS will find all occurrence of the text.

```
USAGE SHOW
150 ...SHOW "PROGRAM CHAINS TO CUSTOMER ENTRY SCREEN"
```

You need to replace the SHOW and WAIT statements (lines 150 and 160, and line 180) with CHAIN statements. (The CHAIN statement is explained in Step 3.) We will enter the following line at the program work area command line.

```
150 CHAIN"CUST_ENTRY"
```

If you make a mistake when entering the text of a program line, move the cursor to the beginning of the line and press the ERASE EOF key. When you press ENTER, the original line will be returned.

Since we are removing the SHOW statement, we no longer need the associated WAIT statement. Enter 160 and press ENTER to remove line 160.

```
160
```

Enter the following at the command line of program work area to replace line 180.

```
180 CHAIN"CUST_BROWSE"
```

Step 3: Understanding the CHAIN statement

The CHAIN statements that you just added to the menu program (lines 150 and 180) replace the program that is currently executing with the program specified in the CHAIN statement, and begin executing that new program:

```
150 ...CHAIN"CUST_ENTRY"
...
180 ...CHAIN"CUST_BROWSE"
```

Step 4: Renumbering your program

Now, follow these steps to renumber your program:

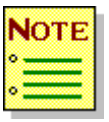
1. Enter the following command on the command line:

   ```
   SEQUENCE 100, 10
   ```

2. Press ENTER.

MANTIS sequences your program line numbers, beginning at line 100 and incrementing the line numbers by ten. Your program should look like this:

```
100 ENTRY CUST_MENU
110 .SCREEN MAP("CUST_MENU")
120 .| THIS IS THE CUSTOMER ACCOUNTS MENU
130 .CONVERSE MAP
140 .WHILE MAP<>"CANCEL"
150 ..WHEN ACTION=1 OR MAP="PF1"
160 ...CHAIN"CUST_ENTRY"
170 ..WHEN ACTION=2 OR MAP="PF2"
180 ...CHAIN"CUST_BROWSE"
190 ..END
200 ..CLEAR MAP
210 ..CONVERSE MAP
220 .END
230 EXIT
```



Don’t run the menu program before replacing it, or your changes will be lost when the CUST_MENU chains to CUST_ENTRY or CUST_BROWSE.

Step 5: Saving your changes

Before running the program, you must save your changes using the REPLACE command. You don't need to specify the program name or password; simply enter REPLACE at the command line and press ENTER. The REPLACE command copies the program in your work area over the existing version in your library.

Exercises

Before proceeding to the next chapter, create the program stubs for the two options on the main Burrys menu. A *stub* is a skeleton program containing temporary code used to execute a simple routine (for example, CONVERSE a screen) and then return to the higher level program.

You will create two stubs, one for the customer entry program (CUST_ENTRY), and one for the customer browse program (CUST_BROWSE). Later, you will replace the stubs with actual routines.

Our versions of the program stubs appear in "Exercise examples" on page 189.

Exercise 1: Creating a stub for the customer entry program

Using the Program Design, create a program stub for the customer entry program, naming it CUST_ENTRY.

This program should display the Burrys new customer screen (CUST_ENTRY) and prepare it to pass data to the program (hint: refer to your ADD_CUST program from an earlier exercise). In addition, you must provide a way to exit from the customer entry program and return to the menu program.

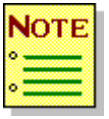
All of the statements you need to complete the CUST_ENTRY stub appear in the menu program.

Be sure to save CUST_ENTRY before you run it.

If you design your program stub correctly, you should be able to display the new customer screen and move from it to the menu screen and then back again. When you've finished entering the stub for CUST_ENTRY, save the program.

Exercise 2: Creating a stub for the browse program

Create a similar stub for the customer browse screen. Call it CUST_BROWSE and provide a way to exit from the program stub and return to the menu program.



Be sure to save CUST_BROWSE before you run it.

Exercise 3: Running the menu program

If the CUST_MENU program is not currently in your work area, enter LOAD"CUST_MENU" on the command line and press ENTER to load the program into the work area.

Now, run the CUST_MENU program by entering RUN at the command line and pressing ENTER. When the menu screen displays, enter “1” in the ACTION field and press ENTER.

MANTIS chains to the stub entry program and displays the customer entry screen. Press CANCEL to return from this screen to the menu. You can now enter 2 in the ACTION field and press ENTER to chain to the stub browse program and display the customer browse screen.

Press CANCEL to return to the menu screen; then, press CANCEL again to return from the menu screen to Program Design.

7

Creating a browse program

In this chapter, you'll expand your knowledge of MANTIS programs, and write a browse program that reads records from a file and displays them on a screen. To do so, you will modify the stub for the CUST_BROWSE program that you created in chapter 6.

Learning outline

In this chapter you will learn how to:

- Understand how automatic mapping can reduce the CPU usage and memory requirements of your program
- Specify a MANTIS internal file that your program will access
- Retrieve a record from a MANTIS file
- Access help in Program Design
- Retrieve a series of records from a MANTIS file and display them on a screen
- Use a counter in a MANTIS program

Basic concepts: Understanding MANTIS file access

In order to read and write records to a MANTIS file, you must use the FILE statement to identify the file that your program will access:

FILE *name1*([*library1:*]*file-name1,password1*[,PREFIX][,*n1*])

[,*name2*([*library2:*]*file-name2,password2*[,PREFIX][,*n2*]) . . .]

In the FILE statement:

- **name*n.*** Specifies a symbolic name for the file that you use in subsequent GET, UPDATE, INSERT, and DELETE statements.
- **library*n.*** (Optional) Specifies the name of the library that contains the file. If the file is in another user's library, you can access it by specifying the name of the user library where it resides. If this parameter is used, the colon (:) is required.
- **file-name*n.*** Specifies the name of the file as it was saved during file design.
- **password*n.*** Specifies the password that indicates the type of file access your program needs (read only, update, or insert/delete).
- **PREFIX.** (Optional) Indicates that MANTIS should place the symbolic name and an underscore before all field names defined in this file view.
- ***n*.** (Optional) Indicates how many buffers (or LEVELS) MANTIS should allocate to this file.

When you run a program that contains a FILE statement, MANTIS retrieves the file design layout from your library and places it in your work area.

Before you create the browse program, it's important to understand how MANTIS allocates memory for SCREEN and FILE statements. The customer browse screen (CUST_BROWSE) contains the following fields:

```
CUST_NUMBER
CUST_NAME
CUST_BRCH_NUMBER
CUST_CREDIT_RAT
CUST_CREDIT_LIM
CUST_COMMENT
MESSAGE
```

Each field except MESSAGE occurs 15 times on the screen, due to its vertical repeat specification. When you issue a SCREEN statement, MANTIS allocates memory locations for 15 occurrences of each field. We call these *arrays*.

To distinguish between one occurrence of a field and another, MANTIS adds subscripts from 1 to 15 to each occurrence:

```
CUST_NUMBER(1)      CUST_COMMENTS(1)
CUST_NUMBER(2)      CUST_COMMENTS(2)
...                 ...
CUST_NUMBER(15)     CUST_COMMENTS(15)
```

The Burrys customer file contains elements that have the same names as the browse screen, plus these additional elements:

```
CUST_ADDRESS
CUST_STATE
CUST_CLASS
CUST_CITY
CUST_ZIP_CODE
```

When you use a FILE statement after a SCREEN statement, MANTIS uses the arrays already allocated by SCREEN, and adds only those variables necessary for symbolic names that are being defined by the file (that is, the five elements listed above).

Automatic mapping is the process that MANTIS uses to allow the sharing of data areas between variables of like name and data type. This mapping occurs automatically whenever an already defined name is encountered within a file or screen design. We recommend that you use standard naming conventions on a system-wide basis to make the best use of this feature and avoid moving values from one variable to another.

If you have programs with large numbers of LET statements to move fields between different areas (for example, screens and files), using automatic mapping can reduce the complexity and chance for errors in your program, and also reduce the CPU usage and memory requirements of your program.

With many languages such as COBOL, the programmer defines the memory locations for screen and file variables separately. When data is read, moved, and written, there can be a considerable use of memory and CPU usage. This is not the case with MANTIS. Automatic mapping saves coding time, memory, and CPU usage.

The advantages of using automatic mapping can be lessened if standard naming conventions are not used throughout the shop. Since MANTIS does duplicate definition checking only in certain situations (such as data type), naming conventions should also include standards for text lengths. If these conventions are not followed, it is possible to have a field defined with two different lengths.

For example, it is possible to define a text field as 55 bytes long on a screen, and 35 bytes long in a file. MANTIS uses the first field length encountered when the program is run. That is, if MANTIS encounters the FILE statement first, the data work area is set up with a length of 35 bytes for the field.

Shops that are database-driven should always specify ACCESS, VIEW, TOTAL, and FILE statements first in a MANTIS program to ensure that the MANTIS program always uses the most current definition for a field.

Step-by-step: Writing a browse program

In this section, you will create a browse program by filling out the browse stub (CUST_BROWSE) that you created in chapter 6. The browse program will read the records in the Burrys customer file and display them, 15 at a time, on your browse screen. When MANTIS reaches the end of the file or when you press CANCEL, control will return to the menu program.

We'll start with a program that retrieves one record at a time from the Burrys customer file and displays it on the browse screen. To begin, follow these steps to load the CUST_BROWSE stub:

1. Select the Design a Program option from the Facility Selection menu.
2. Select Update Program from the Program Design Facility menu and press ENTER. The MANTIS programming work area panel displays.
3. Type LOAD CUST_BROWSE at the bottom of the screen and press ENTER.
4. MANTIS loads the program into work area and you are ready to edit the program.

Step 1: Specifying a MANTIS internal file

You will use the FILE statement to identify the file that your program will access. Supply a symbolic name, the file name, and the password. At the command line, enter the following line; then, press ENTER:

```
125 FILE REC("CUST_INFO","PASSWORD")
```

When you press ENTER, MANTIS adds line 125 to your program:

```
100 ENTRY CUST_BROWSE
110 .SCREEN MAP("CUST_BROWSE")
120 .CONVERSE MAP
125 .FILE REC("CUST_INFO","PASSWORD")
130 .WHILE MAP<>"CANCEL"
140 ..CONVERSE MAP
150 .END
160 .CHAIN"CUST_MENU"
170 EXIT
```

When you run the preceding program, MANTIS will retrieve the file design layout from your library and place it in your work area.

Step 2: Retrieving the first record from the file

Next, use the GET statement to retrieve the first record from the customer file. As part of the GET statement, you must supply the file's symbolic name. On the command line, enter the following line and press ENTER:

```
133 GET REC
```

When you press ENTER, MANTIS adds the GET statement to your program:

```
100 ENTRY CUST_BROWSE
110 .SCREEN MAP("CUST_BROWSE")
120 .CONVERSE MAP
125 .FILE REC("CUST_INFO","PASSWORD")
130 .WHILE MAP<>"CANCEL"
133 ..GET REC
140 ..CONVERSE MAP
150 .END
160 .CHAIN"CUST_MENU"
170 EXIT
```

The GET statement retrieves a record from a file and returns a text value that reflects the status of the GET action. The values are:

Status indicator	Status
FOUND	MANTIS found the record that you requested.
NOTFOUND	The record that you requested does not exist.
NEXT	You issued a GET statement without a key, and MANTIS retrieved the next record in the file.
END	MANTIS failed to retrieve the record because it reached the end of the file.

MANTIS returns these statuses in the file's symbolic name. In other words, if the GET is successful MANTIS sets the variable REC (your file's symbolic name) equal to "NEXT," since you did not supply a key. (You can think of the variable REC as holding a text value that is the status of the prior operation against it.)

Replace and run CUST_BROWSE. To do so, enter REPLACE and press ENTER; then, enter RUN and press ENTER.

Instead of displaying the customer browse screen, MANTIS returns you to programming mode and displays an error message:

```
FAULT 212: Type or dimensions in profile are inconsistent with an
  existing variable CUST_NUMBER
```

Step 3: Accessing help

When you receive an error message, you can use the HELP command to display message. Simply enter HELP FAULT 212 at the bottom of the screen and press ENTER. The following screen displays:

```
 FAULTS

   212


    Type or dimensions in profile are inconsistent with an existing
    variable.

    Explanation:  During processing of an ACCESS, FILE, INTERFACE,
    SCREEN, or ULTRA statement, variables and arrays are defined in your
    work area according to the profile specified in the corresponding
    Design Facility.  When variables or arrays from different profiles
    have the same name, MANTIS assumes that this is intended to effect
    the automatic transfer of data between entities, e.g., between a
    screen and a file record.  MANTIS accordingly checks that the type
    and dimensions are consistent.

    Action:  Use the DISPLAY ALL command to display the attributes of all
    the variables and arrays in your work area.  This will help you to
    identify the conflicting names or attributes.


aaaa11632: END
```

As the help notes, you will usually receive this message when the dimensions for a screen and a file are different. In this instance, your program instructs MANTIS to retrieve one record from the customer file and display it on the screen. But remember, the browse screen has 15 lines on it.

To coordinate your program and screen, you could remove the repeat specifications from the customer browse screen. To do that, you would return to the Screen Design Facility, use the library functions to fetch the screen CUST_BROWSE, and then use the Update repeat specifications option to change the repeat specifications for each field.

Instead of doing that, we'll expand the CUST_BROWSE program to display 15 records at a time from the CUST_INFO file.

Step 4: Using the LEVEL parameter with the FILE statement

Let's take another look at the FILE statement:

FILE *name1*([*library1:*]*file-name1,password1*[,PREFIX][,*n1*])

[,*name2*([*library2:*]*file-name2,password2*[,PREFIX][,*n2*]) . . .]

You need to use the n1 parameter (that is, LEVEL) in your FILE statement to allocate additional memory for those fields unique to the file.

Since the customer browse screen has 15 lines, you should allocate 15 buffers in your FILE statement. Enter ALTER 125 on the command line, press ENTER, and change the line to read.

```
125 FILE REC("CUST_INFO","PASSWORD",15)
```

To save your changes, enter REPLACE at the command line and press ENTER.

The LEVEL parameter also synchronizes by subscript the elements within a record:

```
CUST_NUMBER(1)      CUST_COMMENTS(1)
CUST_NUMBER(2)      CUST_COMMENTS(2)
...                 ...
CUST_NUMBER(15)     CUST_COMMENTS(15)
```

Thus, each record now contains the correct element at every level, from CUST_NUMBER through CUST_COMMENTS.

Step 5: Clearing the records on the browse screen

The customer browse screen can display up to 15 records at a time. These variables will retain their values until they are set to another value or CLEARed (that is, reset to null or zero). If your customer file contains more than 15 records, you should erase the first 15 records from the screen before you read in the next set.

To do this, add a CLEAR statement after 130. On the command line, enter the following line and press ENTER:

```
131 CLEAR MAP
```

MANTIS adds the line to your program:

```
100 ENTRY CUST_BROWSE
110 .SCREEN MAP("CUST_BROWSE")
120 .CONVERSE MAP
125 .FILE REC("CUST_INFO","PASSWORD",15)
130 .WHILE MAP<>"CANCEL"
131 ..CLEAR MAP
```

Now, as the first step in each execution of the WHILE-END loop, MANTIS will erase all prior customer records from the screen.

Step 6: Adding a counter to the program

MANTIS must keep track of its position as it reads the customer file. To do this, you must use a counter.

The LET statement assigns values to a numeric variable:

[LET] ***v*** $\left[\begin{matrix}(i)\\(i,j)\end{matrix}\right]$ **[ROUNDED(*n*)] = *e1* [,*e2*,*e3* . . .]**

Give the counter called BUFFER a starting value of 1. To do so, enter the following statement on the command line and press ENTER:

```
132 LET BUFFER=1
```

When you set BUFFER equal to 1, you are *initializing* the variable. The LET is also optional; instead, you could just write BUFFER=1. Because the symbolic name BUFFER has not been previously defined, it will become a BIG (numeric) data value.

```
100 ENTRY CUST_BROWSE
110 .SCREEN MAP("CUST_BROWSE")
120 .CONVERSE MAP
125 .FILE REC("CUST_INFO","PASSWORD",15)
130 .WHILE MAP<>"CANCEL"
131 ..CLEAR MAP
132 ..LET BUFFER=1
133 ..GET REC
140 ..CONVERSE MAP
150 .END
160 .CHAIN"CUST_MENU"
170 EXIT
```

Step 7: Reading the file record-by-record

Your next step is to construct a second WHILE-END loop that will perform the following tasks:

- Test for the end of the file or end of the screen occurrences
- Read the file record-by-record, beginning with the second record
- Keep track of the buffers on your screen

Examine the following loop:

```
GET REC LEVEL=BUFFER
WHILE REC<>"END"AND BUFFER<15
BUFFER=BUFFER+1
GET REC LEVEL=BUFFER
END
CONVERSE MAP
```

The first statement (GET REC LEVEL=BUFFER) tells MANTIS to retrieve a record and place it in the array element(s) represented by BUFFER.

The WHILE statement determines that the loop will continue as long as MANTIS hasn't reached the end of the file. It also prevents the counter from exceeding the number of levels you specified in the FILE statement. When the maximum number of levels has been retrieved from the file, the screen is also full of data.

The next statement (BUFFER=BUFFER+1) increments the counter by 1 each time MANTIS executes the loop. Since you originally set BUFFER equal to one in statement 132, this statement increases it to two the first time that MANTIS executes the WHILE loop. The GET statement retrieves a record from the level (or buffer) equal to the counter.

To modify your program, add four lines after line 133, or the line that contains your GET REC statement.

Change line 133 and add lines 134 through 137 as shown.

```
133 ..GET REC LEVEL=BUFFER
134       WHILE REC<>"END"AND BUFFERS<15
135       BUFFER=BUFFER+1
136       GET REC LEVEL=BUFFER
137       END
```

Step 8: Terminating the browse

If you were to run your program now, MANTIS would display the file, 15 records at a time, on your screen. But as the program is currently written you cannot return to the menu, even when you reach the end of the file, without pressing CANCEL.

You need to add a condition check that terminates the browse whenever you reach the end of the CUST_INFO file. Modify line 130 to test for both the CANCEL key and the end of file, as shown:

```
100 ENTRY CUST_BROWSE
110 .SCREEN MAP("CUST_BROWSE")
120 .CONVERSE MAP
125 .FILE REC("CUST_INFO","PASSWORD",15)
130 .WHILE MAP<>"CANCEL"AND REC<>"END"
131 ..CLEAR MAP
132 ..LET BUFFER=1
133 ..GET REC LEVEL=BUFFER
134 ..WHILE REC<>"END"AND BUFFER<15
135 ...BUFFER=BUFFER+1
136 ...GET REC LEVEL=BUFFER
137 ..END
140 ..CONVERSE MAP
150 .END
160 .CHAIN"CUST_MENU"
170 EXIT
```

This program will display the contents of the entire file, 15 records at a time. Once it has displayed the entire file, MANTIS will automatically return you to the menu screen. But, you can also press CANCEL to return to the menu screen before you reach the end of the file. Pressing any key other than CANCEL will display the next set of records.

Step 9: Sequencing and replacing the browse program

Renumber your program by tens, beginning with 100. To do so, enter the following statement on the command line and press ENTER:

```
SEQUENCE 100,10
```

Now, check your program for accuracy against the following listing:

```
100 ENTRY CUST_BROWSE
110 .SCREEN MAP("CUST_BROWSE")
120 .CONVERSE MAP
130 .FILE REC("CUST_INFO","PASSWORD",15)
140 .WHILE MAP<>"CANCEL"AND REC<>"END"
150 ..CLEAR MAP
160 ..LET BUFFER=1
170 ..GET REC LEVEL=BUFFER
180 ..WHILE REC<>"END"AND BUFFER<15
190 ...BUFFER=BUFFER+1
200 ...GET REC LEVEL=BUFFER
210 ..END
220 ..CONVERSE MAP
230 .END
240 .CHAIN"CUST_MENU"
250 EXIT
```

Before you run your program, be sure to replace it. Enter REPLACE on the command line and press ENTER.

Run your program. You can return to the menu screen by pressing CANCEL, no matter how many records the file contains. When you reach the end of the file, MANTIS will automatically return you to the menu screen.

Exercises

Before proceeding to the next chapter, complete the exercises in this section.

All of the statements you need to complete the exercises appear in CUST_BROWSE. Our versions of the completed programs appear in “Exercise examples” on page 189.

Exercise 1: Writing a program to browse the state codes file

Write a browse program that reads the state codes file (STATE_CODES) and displays one record at a time on the state code entry screen (STATE_CODE). (Hint: You will not need to use record buffers because only one record at a time will be displayed on the screen.)

The program should read the records in the state codes file until MANTIS reaches the end of the file, or you press CANCEL. Save your program, naming it CODE_BROWSE.

Exercise 2: Enhancing the state codes browse program

Enhance your state code browse program to display as many state codes as your screen can hold. This means that you must add the necessary repeats for the data field, CUST_STATE, to the state code screen. (Our screen can hold 14 codes, which means we added 13 repeat specifications.)

Design your program to chain to the menu screen when MANTIS reaches the end of the file, or you press CANCEL.

8

Creating a data entry program

Next, you'll use Program Design to expand the customer entry stub (CUST_ENTRY) that you wrote in chapter 6, to create a data entry program for the Burrys application.

Learning outline

In this chapter you will learn how to:

- Understand MANTIS program control capabilities
- Write a program that accepts your entries and passes control to a subroutine composed of edit checks
- Display a prompter when the user presses a PF key
- Write a subroutine that will perform the following tests on the customer information you provide:
  - Verify that the CUST_NAME field contains an entry
  - Verify the accuracy of the state code that is entered in the CUST_STATE field
  - Verify that the entry in the CUST_NUMBER field contains six characters
  - Check CUST_NUMBER to see if it already exists in the customer file
- Use the ATTRIBUTE statement to alter the attributes of a data field on an existing screen design

Basic concepts: Understanding program control basics

This section discusses MANTIS program control capabilities, and explains how to code the DO statement to pass control to a program subroutine.

Program control capabilities

MANTIS provides the following program control capabilities:

- **CHAIN.** Allows a MANTIS program to transfer data and control of execution to another MANTIS program without an automatic return path. (You used the CHAIN statement in chapter 6, when you created the CUST_MENU program.)

- **DO.** There are two types of DO statements in MANTIS. *Internal* DO allows a MANTIS program to transfer data and control of execution to a routine within the same program, and return to the next statement following the DO. *External* DO allows a MANTIS program to transfer data and control of execution to another MANTIS program with an automatic return path to the next statement. (You will learn how to code both types of DO statements later in this chapter.)

- **CALL.** Allows a MANTIS program to transfer data and control of execution to a non-MANTIS program and automatically return to the next statement following the CALL. (For more information on using the CALL statement, refer to AD/Advantage MANTIS Language OpenVMS/UNIX, P39-1310.)

- **PERFORM.** Allows a MANTIS program to transfer control of execution to a non-MANTIS program and automatically return to the next statement. The PERFORM statement also has options to transfer control without return and to execute background tasks. (For more information on using the PERFORM statement, refer to AD/Advantage MANTIS Language OpenVMS/UNIX, P39-1310.)

Using the DO statement

The DO statement transfers program execution to an internal or external subroutine. A *subroutine* is a block of statements either within the existing MANTIS program, or identified by a PROGRAM statement, that performs a function required at one or more points in a program.

Internal DO transfers data and control of execution to a routine *within the same program*, and returns to the next statement following the DO. External DO transfers data and control of execution to *another MANTIS program*, with an automatic return path to the next statement following the DO.

Any program can use both internal and external DO statements. Internal routines are defined by ENTRY statements within the calling program. External routines are defined in PROGRAM statements within the calling program.

External DO allows multiple programs to share common subroutines. Using external DO, you can create programs that can be shared by other users, providing access to both internal and external subroutines. You can also link subroutines with external DO.

Use the following syntax to code the DO statement:

DO *entry-name*[(*argument1*,*argument2*,...)]

In the DO statement:

- **entry-name.** Specifies the name of a subroutine as indicated in the ENTRY or PROGRAM statement.
- **argument.** (Optional) Specifies the argument(s) you want passed to the subroutine. A maximum of 255 arguments can be passed in one DO statement.

When you code the DO statement, it must appear on a line by itself. Any additional statements coded on the end of a DO statement (and separated with a colon) are ignored by MANTIS.

You must include an ENTRY-EXIT statement around the subroutine that you specify in the DO statement.

All data names (arguments) must be defined before they are used on a DO statement. Only variables passed as arguments to an external subroutine are available to the external routine. If the variable is a SCREEN, FILE, VIEW, TOTAL, ACCESS, or INTERFACE variable, sub variables are not available unless explicitly passed.

Before you can use the DO statement to invoke an *external* subroutine, you must code the PROGRAM statement to identify the external subroutine to MANTIS.

Use the following syntax to code the PROGRAM statement:

PROGRAM *name1*([*library1*:]*program-name1*,*password1*)
[, *name2*([*library2*:]*program-name2*,*password2*) . . .]

In the PROGRAM statement:

- **name*n*.** Specifies the symbolic name you will use to refer to your program in subsequent DO statements.
- **[library*n*:]program-name*n*.** Specifies the name of the program as you saved it in program design.
- **password.** Specifies the password as you saved it during program design.

The following program calls both an external subroutine (EDIT_RTN, lines 40 and 90) and an internal subroutine (ERROR_RTN, lines 60 and 110):

```
10 ENTRY EDIT_PROGRAM
20 .TYPE="CREDIT CHECK"
30 .PROGRAM EDIT_RTN("VALIDATION","COMMON")
40 .DO EDIT_RTN(TYPE,CUST_NO,STATUS,MESSAGE)
50 .IF STATUS<>"GOOD"
60 ..DO ERROR_RTN(CUST_NO)
70 .END
80 .TYPE="SELECT SALES REP"
90 .DO EDIT_RTN(TYPE,CUST_NO,STATUS,MESSAGE)
100 .IF STATUS<>"GOOD"
110 ..DO ERROR_RTN(SALES_REP)
120 .ELSE
130 ..SALES_REP=MESSAGE
140 .END
150 EXIT
160 ENTRY ERROR_RTN(FIELD)
170 .IF NOTE=" "
180 ..NOTE=MESSAGE
190 ..ATTRIBUTE(MAP,FIELD)="BRI,CUR"
200 .ELSE
210 ..ATTRIBUTE(MAP,FIELD)="BRI"
220 .END
230 EXIT
```

If used optimally, external DO can improve the efficiency of program execution and make it easier to maintain your applications. However, there are many considerations for using external DO effectively that are beyond the scope of the examples provided in this tutorial. For more detailed information on using external DO, refer to *AD/Advantage MANTIS Language OpenVMS/UNIX*, P39-1310.

Step-by-step: Creating the customer entry program

You will create the insert module first, and then design your edit checks. To begin, sign on to MANTIS and select the Design a Program option from the Facility Selection menu. On the Program Design Facility command line, enter 1 and press ENTER. The Program Directory List displays:

```
PRG003  BURRYS                Program Directory Selection        2001/01/03
                                                                   14:28:11
 SEL------------NAME------------- -----STATUS----- --------DESCRIPTION-------->>
___ADD_CUST
 __CODE_BROWSE
 __CUST_BROWSE
 s_CUST_ENTRY
 __CUST_MENU
 __STATE_CODES
```

Enter an s on the blank line next to CUST_ENTRY; then, press ENTER. MANTIS displays the program stub that you created in chapter 6.

Your completed program must perform these steps:

1. Define the customer information file
2. Insert the record into the file
3. Clear your entry from the screen and CONVERSE the screen again
4. Call the edit subroutine

After you create the insert module, you will then create a subroutine to edit the data as it is entered.

Step 1: Defining the file

Begin by adding a FILE statement to the program stub (CUST_ENTRY) that you developed earlier. Since you'll insert only one record at a time, you don't need to specify a level. On the command line, enter the following statement and press ENTER:

```
115 FILE REC("CUST_INFO","PASSWORD")
```

When you press ENTER, MANTIS adds the line to your program:

```
100 ENTRY CUST_ENTRY
110 .SCREEN MAP("CUST_ENTRY")
115 .FILE REC("CUST_INFO","PASSWORD")
120 .CONVERSE MAP
130 .WHILE MAP<>"CANCEL"
140 ..CONVERSE MAP
150 .END
160 .CHAIN"CUST_MENU"
170 EXIT
```

Step 2: Inserting a record into the file

The INSERT statement adds a record to a MANTIS file. With INSERT, you must specify the file's symbolic name and, if necessary, the level. Since you are not using a level, your INSERT statement should read:

```
134 INSERT REC
```

To add this statement, enter it on the command line and press ENTER.

Step 3: Clearing the entry

After MANTIS inserts the record, you'll want to clear the screen (and thus, all the variables in it) for your next entry (either another customer, or CANCEL). Add the CLEAR statement to your program as shown below:

```
136 CLEAR MAP
```

Replace and run your program. When the new customer screen appears, you can enter customer information and press ENTER:

```
                    B U R R Y S
                NEW CUSTOMER ENTRY


          NAME:
          ADDRESS:
          CITY:
          STATE:
          ZIP CODE:

          CUSTOMER NUMBER:
          CLASS:
          CUSTOMER CREDIT RATING:
          CREDIT LIMIT:
          BRANCH NUMBER:
          COMMENTS:
```

Step 4: Calling the edit subroutine

As it is currently written, your program will insert records into a MANTIS file, but it will not perform any special editing routines on the records you insert. Given the size of the Burrys Corporation, you must provide some basic editing tests to ensure the integrity of the customer file. These edit checks will perform four tests on the data for each new customer that a user enters:

- Verify that the CUST_NAME field contains an entry
- Verify the accuracy of the state code that is entered in the CUST_STATE field
- Verify that the entry in the CUST_NUMBER field contains six characters
- Check CUST_NUMBER to see if it already exists in the customer file

Since the subroutine will check your entries for errors, you should begin by declaring the variable ERROR and setting it equal to FALSE:

```
131  SMALL ERROR:ERROR=FALSE
```

FALSE is a MANTIS numeric built-in function that evaluates to 0. Use a numeric variable to hold the Boolean values FALSE (0) and TRUE (1).

Next, you must instruct MANTIS to execute the subroutine. The DO statement transfers program control from a program's current level to a subroutine. Add the following program line and assign a symbolic name to your subroutine in the DO statement as shown:

```
132 DO VALIDATE_CUST_INFO
```

Now, you need to define the edit subroutine. Add the following statements to your program:

```
300 ENTRY VALIDATE_CUST_INFO
310 .SHOW"VALIDATE_CUST_INFO":WAIT
320 EXIT
```

When MANTIS executes the DO statement in line 132, it will pass control to the edit subroutine in line 300. Then, when MANTIS executes the EXIT statement in line 320, it will return control to the statement following the DO statement (in your program, line 133).

Step 5: Using the IF-END structure to test for errors

To complete the insert (main line) portion of the program, add an IF-END structure that will insert your record if the subroutine returns a value of FALSE for the variable ERROR. Add these statements to your program as shown:

```
133 IF NOT(ERROR)
137 END
```

The NOT function returns TRUE (1) for the expression in line 133 if ERROR evaluates to FALSE (0); otherwise, the NOT function returns FALSE (0).

The IF-END structure executes a block of statements only if the condition you specify is true (in our program, if NOT(ERROR). Otherwise, MANTIS ignores the block of statements between IF and END. When you include an ELSE statement (IF-ELSE-END) and the condition you specify is false, MANTIS executes the block of statements after ELSE.

Next, add a blank comment on line 180 to make it easier to see the subroutine. Your program should now look like this:

```
100 ENTRY CUST_ENTRY
110 .SCREEN MAP("CUST_ENTRY")
115 .FILE REC("CUST_INFO","PASSWORD")
120 .CONVERSE MAP
130 .WHILE MAP<>"CANCEL"
131 ..SMALL ERROR:ERROR=FALSE
132 ..DO VALIDATE_CUST_INFO
133 ..IF NOT(ERROR)
134 ...INSERT REC
136 ...CLEAR MAP
137 ..END
140 ..CONVERSE MAP
150 .END
160 .CHAIN"CUST_MENU"
170 EXIT
180 |
300 ENTRY VALIDATE_CUST_INFO
310 .SHOW"VALIDATE_CUST_INFO":WAIT
320 EXIT
```

Step 6: Testing your program structure

To test your program, temporarily add a vertical bar in front of the CHAIN statement on line 160:

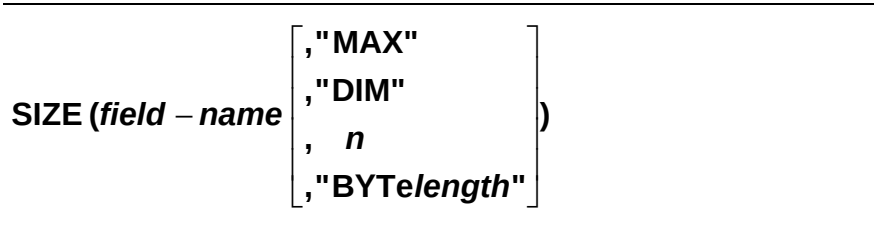
```
160 |CHAIN"CUST_MENU"
```

Adding the vertical bar changes this line into a comment, so you can run your program without chaining to the CUST_MENU program. Save your changes by entering REPLACE on the command line and pressing ENTER, then run the program.

When the CUST_ENTRY screen displays, enter a customer number and press ENTER. MANTIS displays the text string VALIDATE_CUST_INFO that follows the SHOW statement (line 310), indicating that MANTIS has successfully passed control to the edit subroutine. Press CANCEL to return to the MANTIS Program Work Area.

Step 7: Adding an edit routine to test for an entry in a field

Your first edit check will test the CUST_NAME field for an entry. If an operator leaves the text field CUST_NAME blank, MANTIS sets the field to zero length (NULL). You will use the SIZE function to test for this condition:



The SIZE function returns the maximum or current length of a field. It can also return the number of defined dimensions for a field or array, as well as the number of occurrences for a specific dimension of an array. In this case, we'll use the default option for returning the current length of the data.

When using this function in your program, you must supply the field name in parentheses:

```
IF SIZE(CUST_NAME)=0
```

If CUST_NAME equals zero, set the variable, ERROR (initialized at line 131), equal to TRUE:

```
IF SIZE(CUST_NAME)=0
ERROR=TRUE
```

TRUE is a MANTIS numeric built-in function that evaluates to +1.

Your customer entry screen contains a field named MESSAGE that you can use to return messages to users. If you set MESSAGE equal to a text literal in your program, MANTIS will display the message on the screen:

```
IF SIZE(CUST_NAME)=0
ERROR=TRUE
MESSAGE="CUSTOMER NAME MISSING-–PLEASE ENTER"
```

Remember to enclose your text literal in double quotes " ".

In addition to returning an error message, you can use the ATTRIBUTE statement to highlight the field in error, and place the cursor at the beginning of the field. The ATTRIBUTE statement alters the attributes of a data field on an existing screen design.

After ATTRIBUTE, supply the screen and field names enclosed in parentheses and separated by a comma. To place the cursor in the field, set ATTRIBUTE equal to CUR (an abbreviation for CURSOR). To highlight the field, add BRI (an abbreviation for BRIGHT) after CUR. Enclose them in double quotes, as shown below. (For more detailed information on using the ATTRIBUTE statement, refer to AD/Advantage MANTIS Language OpenVMS/UNIX, P39-1310.)

ALTER line 310 with the SIZE function. Then, add the following statements after line 310 to your program, so that it looks like this:

```
310 IF SIZE(CUST_NAME)=0
311 .ERROR=TRUE
312 .ATTRIBUTE(MAP,CUST_NAME)="CUR,BRI"
313 .MESSAGE="CUSTOMER NAME MISSING--PLEASE ENTER"
314 END
```

Now, renumber your program to leave room for additional statements. To do so, at the command line enter the following command and press ENTER:

```
SEQUENCE 100,10
```

Your program should look like this:

```
100 ENTRY CUST_ENTRY
110 .SCREEN MAP("CUST_ENTRY")
120 .FILE REC("CUST_INFO","PASSWORD")
130 .CONVERSE MAP
140 .WHILE MAP<>"CANCEL"
150 ..SMALL ERROR:ERROR=FALSE
160 ..DO VALIDATE_CUST_INFO
170 ..IF NOT(ERROR)
180 ...INSERT REC
190 ...CLEAR MAP
200 ..END
210 ..CONVERSE MAP
220 .END
230 |CHAIN"CUST_MENU"
240 EXIT
250 |
260 ENTRY VALIDATE_CUST_INFO
270 .IF SIZE(CUST_NAME)=0
280 ..ERROR=TRUE
290 ..ATTRIBUTE(MAP,CUST_NAME)="CUR,BRI"
300 ..MESSAGE="CUSTOMER NAME MISSING-–PLEASE ENTER"
310 .END
320 EXIT
```

Remove the comment bar before the CHAIN statement on line 230, then enter the REPLACE command to replace your program. Then, enter the RUN command to run the program.

Press ENTER without keying in a customer name. MANTIS returns your error message at the bottom of the new customer screen, and places the cursor in the first position of the CUST_NAME field:

```
                              B U R R Y S
                          NEW CUSTOMER ENTRY


               NAME:                        _
               ADDRESS:
               CITY:
               STATE:
               ZIP CODE:

               CUSTOMER NUMBER:
               CLASS:
               CUSTOMER CREDIT RATING:
               CREDIT LIMIT:
               BRANCH NUMBER:
               COMMENTS:

CUSTOMER NAME MISSING—PLEASE ENTER
```

If you enter a customer name and customer number and press ENTER, MANTIS inserts that record into the Burrys customer file, clears the screen, and waits for your next entry.

Press CANCEL to return to the menu screen. Press CANCEL again to return to MANTIS Program Design.

At the command line, enter LOAD CUST_ENTRY; then, press ENTER. MANTIS loads the program. Enter LIST; then, press ENTER, MANTIS lists the program:

```
LOAD CUST_ENTRY
LIST
100 ENTRY CUST_ENTRY
110 .SCREEN MAP(“CUST_ENTRY”)
120 .FILE REC(“CUST_INFO”,“PASSWORD”)
130 .CONVERSE MAP
140 .WHILE MAP<>”CANCEL”
150 ..SMALL ERROR:ERROR=FALSE
160 ..DO VALIDATE_CUST_INFO
170 ..IF NOT(ERROR)
180 ...INSERT REC
190 ...CLEAR MAP
200 ..END
210 ..CONVERSE MAP
220 .END
230 .CHAIN“CUST_MENU”
240 EXIT
250 |
260 ENTRY VALIDATE_CUST_INFO
270 .IF SIZE(CUST_NAME)=0
280 ..ERROR=TRUE
290 ..ATTRIBUTE(MAP,CUST_NAME)=“CUR,BRI”
300 ..MESSAGE=“CUSTOMER NAME MISSING--PLEASE ENTER”                MORE
```

Notice that the screen can no longer show the entire program. MANTIS shows only the first 21 lines of the CUST_ENTRY program. To view the remainder of the program, press ENTER.

Step 8: Adding an edit routine to test the number of characters entered

The second edit check also uses the SIZE function, this time to test the CUST_NUMBER field. The screen will not accept more than six characters, but an operator might inadvertently enter fewer than six characters.

Use the SIZE function to check CUST_NUMBER for fewer than six characters:

```
IF SIZE(CUST_NUMBER)<6
```

Remember that the SIZE function returns the current size of the field. The < symbol is a MANTIS operator that means “is less than.” Otherwise, this test is similar in structure to the first one.

If your IF statement evaluates to true, then MANTIS should set ERROR equal to TRUE and return an appropriate message to the user. Also, MANTIS should place the cursor at the beginning of the field and highlight it:

```
IF SIZE(CUST_NUMBER)<6
ERROR=TRUE
ATTRIBUTE(MAP,CUST_NUMBER)="CUR,BRI"
MESSAGE="CUSTOMER NUMBER MUST BE 6 CHARACTERS"
END
```

Add these statements as part of an ELSE block after your first edit check (that is, the IF block beginning at statement 270):

```
IF expression
   block a (first edit check)
ELSE
   block b (second edit check)
END
```

This means that you must change statement 310 from END to ELSE, add the second IF clause, and add an additional END statement to terminate the original IF clause.

Type in 6 lines of statements after line 310. Your subroutine should look like this:

```
260 ENTRY VALIDATE_CUST_INFO
270 .IF SIZE(CUST_NAME)=0
280 ..ERROR=TRUE
290 ..ATTRIBUTE(MAP,CUST_NAME)="CUR,BRI"
300 ..MESSAGE="CUSTOMER NAME MISSING--PLEASE ENTER"
310 .ELSE
311 ..IF SIZE(CUST_NUMBER)<6
312 ...ERROR=TRUE
313 ...ATTRIBUTE(MAP,CUST_NUMBER)="CUR,BRI"
314 ...MESSAGE="CUSTOMER NUMBER MUST BE 6 CHARACTERS"
315 ..END
316 .END
320 EXIT
```

Notice the program flow in your subroutine. If the SIZE function in statement 270 returns a value of zero, MANTIS executes lines 280-300. Otherwise, MANTIS ignores lines 280-300 and executes line 311.

If the comparison in line 311 returns a value of TRUE, MANTIS executes lines 312-315. If line 311 evaluates to FALSE, MANTIS executes line 320, and control returns to statement 170, IFNOT(ERROR).

After you correct an error on the new customer screen, MANTIS executes the subroutine again from statement 270.

Now, replace and run the program; then, press ENTER without entering a customer name. Again, the message displays asking you to enter a customer name:

```
                              B U R R Y S
                          NEW CUSTOMER ENTRY


                NAME:                          _
                ADDRESS:
                CITY:
                STATE:
                ZIP CODE:

                CUSTOMER NUMBER:
                CLASS:
                CUSTOMER CREDIT RATING:
                CREDIT LIMIT:
                BRANCH NUMBER:
                COMMENTS:

CUSTOMER NAME MISSING—PLEASE ENTER
```

Enter a customer name and 3 digits of the customer number; then, press ENTER. MANTIS executes the second edit check and highlights CUST_NUMBER:

```
                              B U R R Y S
                          NEW CUSTOMER ENTRY


                NAME:                          Customer Name
                ADDRESS:
                CITY:
                STATE:
                ZIP CODE:

                CUSTOMER NUMBER:               123
                CLASS:
                CUSTOMER CREDIT RATING:
                CREDIT LIMIT:
                BRANCH NUMBER:
                COMMENTS:

CUSTOMER NUMBER MUST BE 6 CHARACTERS
```

Notice, however, that both fields remain highlighted. That's because there is nothing in your program to reset the attributes for CUST_NAME.

To avoid highlighting more than one field at a time, use the ATTRIBUTE statement to RESET field attributes after each execution of the subroutine. Press CANCEL to exit from the customer entry screen; then, press CANCEL again to exit from the main menu and display the menu program in the Program Design area.

Load CUST_ENTRY, then enter the following statement on the command line and press ENTER:

```
215 ATTRIBUTE(MAP)="RES"
```

This statement tells MANTIS to reset the attributes to their originally defined value (from the SCREEN design) for all the fields on your screen. Use the SEQUENCE command to renumber your program, then replace it and run it again, stepping through the error messages using the procedure that you followed above. This time, when MANTIS returns the second error message, only the CUST_NUMBER field is highlighted.

Step 9: Adding an edit routine to validate the state code

The third edit check will test the state code that you enter against the state codes file you set up earlier. If MANTIS doesn't find a record in the state codes file that is identical to your screen entry, then your state code is either missing or incorrect.

You must add two statements to your program before creating the edit routine itself. First, identify and name the state codes file. Supply a unique symbolic name, the library name, and the password:

```
125 FILE REC2("STATE_CODES","PASSWORD")
```

Secondly, you must tell MANTIS to read the file for a record identical to the entry on your screen. For this, you will use the GET statement with a key:

GET *file - name* [**(*key1*, *key2*, . . .)[EQUAL]** | **FIRST** | **NEXT** | **PRIOR** | **LAST**] **[ENQUEUE] [LEVEL = *n*]**

A *key* is any file element that is unique to a record. Since a key is unique to a record, you can use it to identify the particular record for which you want MANTIS to read the file.

Use the file element, CUST_STATE, as the key in your GET statement. Supply the file's symbolic name and enclose the key in parentheses:

```
GET REC2(CUST_STATE)
```

If MANTIS returns any status other than "FOUND" for CUST_STATE, your state code is either missing or incorrect. Thus, the IF statement in your edit check should read:

```
IF REC2<>"FOUND"
```

Otherwise, this edit check is similar to the first two. Add it after line 310, so that MANTIS will edit data fields from the top of the screen to the bottom. Add 6 lines of statements after line 310: Here is the additional code you need:

```
311 GET REC2(CUST_STATE)
312 IF REC2<>"FOUND"
313 ERROR=TRUE
314 ATTRIBUTE(MAP,CUST_STATE)="CUR,BRI"
315 MESSAGE="INCORRECT OR MISSING CODE"
316 ELSE
```

You must also add an additional END statement before the EXIT statement in the subroutine.

After you renumber your program, it should look like this:

```
100 ENTRY CUST_ENTRY
110 .SCREEN MAP("CUST_ENTRY")
120 .FILE REC("CUST_INFO","PASSWORD")
130 .FILE REC2("STATE_CODES","PASSWORD")
140 .CONVERSE MAP
150 .WHILE MAP<>"CANCEL"
160 ..SMALL ERROR:ERROR=FALSE
170 ..DO VALIDATE_CUST_INFO
180 ..IF NOT(ERROR)
190 ...INSERT REC
200 ...CLEAR MAP
210 ..END
220 ..CONVERSE MAP
230 ..ATTRIBUTE(MAP)="RES"
240 .END
250 .CHAIN"CUST_MENU"
260 EXIT
270 |
280 ENTRY VALIDATE_CUST_INFO
290 .IF SIZE(CUST_NAME)=0
300 ..ERROR=TRUE
310 ..ATTRIBUTE(MAP,CUST_NAME)="CUR,BRI"
320 ..MESSAGE="CUSTOMER NAME MISSING-PLEASE ENTER"
330 .ELSE
340 ..GET REC2(CUST_STATE)
350 ..IF REC2<>"FOUND"
360 ...ERROR=TRUE
370 ...ATTRIBUTE(MAP,CUST_STATE)="CUR,BRI"
380 ...MESSAGE="INCORRECT OR MISSING CODE"
390 ..ELSE
400 ...IF SIZE(CUST_NUMBER)<6
410 ....ERROR=TRUE
420 ....ATTRIBUTE(MAP,CUST_NUMBER)="CUR,BRI"
430 ....MESSAGE="CUSTOMER NUMBER MUST BE 6 CHARACTERS"
440 ...END
450 ..END
460 .END
470 EXIT
```

Note that this program stops further checking after the first error is found. We could have used a WHEN structure instead of the nested IF statements to find all errors in one pass. But then, you would need some way to determine which error message(s) are displayed when a number of fields are highlighted.

Replace and run your program. Enter a customer name, then enter AA in the CUST_STATE field and press ENTER. MANTIS returns the error message, but it also overwrites AA with the first record from the state codes file (in our file, AK):

```
                              B U R R Y S
                          NEW CUSTOMER ENTRY


               NAME:                        JACK SMITH
               ADDRESS:
               CITY:
               STATE:                       AK
               ZIP CODE:

               CUSTOMER NUMBER:
               CLASS:
               CUSTOMER CREDIT RATING:
               CREDIT LIMIT:
               BRANCH NUMBER:
               COMMENTS:

INCORRECT OR MISSING CODE
```

When MANTIS reads a file, it always returns a record (whether it's the one you requested, or the next record), unless you specified a key in the GET statement or MANTIS has reached the end of the file. In this example, MANTIS couldn't find the invalid AA, so it returned the next higher record in the file.

That record actually appears on your screen because the screen field and the file's key element are both named CUST_STATE. In other words, when field and element names are identical, MANTIS maps data from the file to the screen. This is another example of *automatic mapping*.

To avoid overwriting data on the screen, you can add the EQUAL parameter to the GET statement. (Another technique used by MANTIS programmers to prevent editing reads from overlaying entered text is to define a second FILE statement with a different symbolic name and use the PREFIX option. This prevents the editing read from automatic mapping to screen fields. Step 11 will present an example of this technique.)

To return directly to the CUST_ENTRY program, use the tab key to move the cursor to the lower right corner of the customer entry screen, then enter KILL and press ENTER. KILL will stop an executing program at the CONVERSE statement.

Now, change line 340 as shown:

```
340 GET REC2(CUST_STATE)EQUAL
```

The EQUAL option tells MANTIS to retrieve a record that has an exact key match to your request.

Use the EQUAL parameter only with a keyed GET.

Replace your program and run it. Supply a customer name, and enter AA in the State field. This time, MANTIS returns the error message without retrieving the next record in the state codes file:

```
                              B U R R Y S
                          NEW CUSTOMER ENTRY


                NAME:                          JACK SMITH
                ADDRESS:
                CITY:
                STATE:                         AA
                ZIP CODE:

                CUSTOMER NUMBER:
                CLASS:
                CUSTOMER CREDIT RATING:
                CREDIT LIMIT:
                BRANCH NUMBER:
                COMMENTS:

INCORRECT OR MISSING CODE
```

Step 10: Displaying the state codes prompter

You can use the PROMPT statement to display informational screens such as the state codes prompter. Supply the library name of the prompter, and enclose it in quotes. Use a WHEN-END structure to show the prompter if the user presses PF1. Enter statements 215-217 as shown:

```
215 WHEN MAP="PF1"
216 PROMPT"STATE_CODES"
217 END
```

Before you run the program, enhance the error message on line 380:

```
380 MESSAGE="INCORRECT OR MISSING CODE—PRESS 'PF1' FOR HELP"
```

When you enter the previous line, remember to use single quotes (') around PF1 in the message. Alternately, you could use two pairs of double quotes (") around PF1 ("PRESS ""PF1"" FOR HELP").

However, you could *not* use a single pair of double quotes ("PRESS "PF1" FOR HELP") because double quotes designate a text literal and MANTIS would interpret the message as two text literals. If you enter a single pair of double quotes around PF1, MANTIS will generate an error message when you run your program.

Replace and run the program. Then, press PF1 to view the state codes prompter that you built earlier:

```
                                STATE CODES

AL - ALABAMA                 KY - KENTUCKY              ND - NORTH DAKOTA
AK - ALASKA                  LA - LOUISIANA             OH - OHIO
AZ - ARIZONA                 ME - MAINE                 OK - OKLAHOMA
AR - ARKANSAS                MD - MARYLAND              OR - OREGON
CA - CALIFORNIA              MA - MASSACHUTSETTS        PA - PENNSYLVANIA
CO - COLORADO                MI - MICHIGAN              RI - RHODE ISLAND
CT - CONNECTICUT             MN - MINNESOTA             SC - SOUTH CAROLINA
DE - DELAWARE                MS - MISSISSIPPI           SD - SOUTH DAKOTA
DC - DISTRICT OF COLUMBIA    MO - MISSOURI              TN - TENNESSEE
FL - FLORIDA                 MT - MONTANA               TX - TEXAS
GA - GEORGIA                 NE - NEBRASKA              UT - UTAH
HI - HAWAII                  NV - NEVADA                VT - VERMONT
ID - IDAHO                   NH - NEW HAMPSHIRE         VA - VIRGINIA
IL - ILLINOIS                NJ - NEW JERSEY            WA - WASHINGTON
IN - INDIANA                 NM - NEW MEXICO            WV - WEST VIRGINIA
IA - IOWA                    NY - NEW YORK              WI - WISCONSIN
KS - KANSAS                  NC - NORTH CAROLINA        WY - WYOMING
```

Now, press ENTER to return to the new customer screen:

```
                              B U R R Y S
                           NEW CUSTOMER ENTRY


                  NAME:                          _
                  ADDRESS:
                  CITY:
                  STATE:
                  ZIP CODE:

                  CUSTOMER NUMBER:
                  CLASS:
                  CUSTOMER CREDIT RATING:
                  CREDIT LIMIT:
                  BRANCH NUMBER:
                  COMMENTS:

CUSTOMER NAME MISSING—PLEASE ENTER
```

Notice that MANTIS returns an error message for the CUST_NAME field.

Think for a moment about your program's flow:

```
100 ENTRY CUST_ENTRY
110 .SCREEN MAP("CUST_ENTRY")
120 .FILE REC("CUST_INFO","PASSWORD")
130 .FILE REC2("STATE_CODES","PASSWORD")
140 .CONVERSE MAP
150 .WHILE MAP<>"CANCEL"
160 ..SMALL ERROR:ERROR=FALSE
170 ..DO VALIDATE_CUST_INFO
180 ..IF NOT(ERROR)
190 ...INSERT REC
200 ...CLEAR MAP
210 ..END
215 ..WHEN MAP="PF1"
216 ...PROMPT"STATE_CODES"
217 ..END
220 ..CONVERSE MAP
230 ..ATTRIBUTE(MAP)="RES"
240 .END
250 .CHAIN"CUST_MENU"
260 EXIT
```

The first CONVERSE occurs before the while loop (line 140). When you press PF1 at the first CONVERSE, control enters the WHILE loop and executes the validation routine, which issues the error condition because there is no customer name.

Control then passes to the PROMPT statement, which displays the state codes prompter. When you leave the prompter and CONVERSE the screen again, the error message displays.

You can eliminate this problem by testing for PF1 in your program's main module. To do so, enter the following statement on the command line and press ENTER:

```
155 WHEN MAP<>"PF1"
```

Your program should now look like this:

```
100 ENTRY CUST_ENTRY
110 .SCREEN MAP("CUST_ENTRY")
120 .FILE REC("CUST_INFO","PASSWORD")
130 .FILE REC2("STATE_CODES","PASSWORD")
140 .CONVERSE MAP
150 .WHILE MAP<>"CANCEL"
155 ..WHEN MAP<>"PF1"
160 ...SMALL ERROR:ERROR=FALSE
170 ...DO VALIDATE_CUST_INFO
180 ...IF NOT(ERROR)
190 ....INSERT REC
200 ....CLEAR MAP
210 ...END
215 ..WHEN MAP="PF1"
216 ...PROMPT"STATE_CODES"
217 ..END
220 ..CONVERSE MAP
230 ..ATTRIBUTE(MAP)="RES"
240 .END
250 .CHAIN"CUST_MENU"
260 EXIT
```

Replace and run the program again; then, press PF1 to view the prompter. When you return from the prompter to the new customer screen, MANTIS will not display an error message.

Step 11: Adding an edit routine to test for a duplicate entry

The last edit routine tests the customer number to see if it already exists in the Burrys customer file. This edit check is a little more complicated than the first three.

To test for a duplicate customer number, you must tell MANTIS to read the file for a customer number identical to your screen entry. But, if MANTIS finds such a customer number in the file, it retrieves the record and displays it on your screen. In other words, MANTIS will overwrite your screen entry with the record which already contains the customer number you provided.

You can eliminate this problem by using the PREFIX option in your FILE statement. Add the following file statement to your program, supplying a second symbolic name for the Burrys file and specifying PREFIX as shown:

```
135 .FILE RECX("CUST_INFO","PASSWORD",PREFIX)
```

PREFIX tells MANTIS to place the symbolic name (in this case, RECX) and an underscore before all field names associated with the file design. Where MANTIS calls the customer number in the unprefixed data area CUST_NUMBER, it calls the same field in the prefixed data area RECX_CUST_NUMBER. In effect, MANTIS sets up a second data area for the Burrys customer file.

This second data area contains independent information as the first data area, and with different names. The second, prefixed, variables can be used in this edit check without fear of overlaying the screen input data with data from the file when the record is read.

If MANTIS doesn't find the customer number in the prefixed file, it is unique and you can enter it into the customer file.

Otherwise, this edit check is similar to the others you've written. Add these statements as an ELSE block after the existing edit checks. This means you must add another END statement before EXIT:

```
431 ELSE
432 GET RECX(CUST_NUMBER)
433 IF RECX="FOUND"
434 ERROR=TRUE
435 ATTRIBUTE(MAP,CUST_NUMBER)="CUR,BRI"
436 MESSAGE="DUPLICATE CUSTOMER NUMBER--PLEASE CHANGE"
437 END
```

Renumber and replace your program, then compare it with this listing:

```
100 ENTRY CUST_ENTRY
110 .SCREEN MAP("CUST_ENTRY")
120 .FILE REC("CUST_INFO","PASSWORD")
130 .FILE REC2("STATE_CODES","PASSWORD")
140 .FILE RECX("CUST_INFO","PASSWORD",PREFIX)
150 .CONVERSE MAP
160 .WHILE MAP<>"CANCEL"
170 ..WHEN MAP<>"PF1"
180 ...SMALL ERROR:ERROR=FALSE
190 ...DO VALIDATE_CUST_INFO
200 ...IF NOT(ERROR)
210 ....INSERT REC
220 ....CLEAR MAP
230 ...END
240 ..WHEN MAP="PF1"
250    PROMPT"STATE_CODES"
260 ..END
270 ..CONVERSE MAP
280 ..ATTRIBUTE(MAP)="RES"
290 .END
300 .CHAIN"CUST_MENU"
310 EXIT
320 |
330 ENTRY VALIDATE_CUST_INFO
340 .IF SIZE(CUST_NAME)=0
350 ..ERROR=TRUE
360 ..ATTRIBUTE(MAP,CUST_NAME)="CUR,BRI"
370 ..MESSAGE="CUSTOMER NAME MISSING--PLEASE ENTER"
380 .ELSE
390 ..GET REC2(CUST_STATE)EQUAL
400 ..IF REC2<>"FOUND"
410 ...ERROR=TRUE
420 ...ATTRIBUTE(MAP,CUST_STATE)="CUR,BRI"
430 ...MESSAGE="INCORRECT OR MISSING CODE--PRESS 'PF1' FOR HELP"
440 ..ELSE
450 ...IF SIZE(CUST_NUMBER)<6
460 ....ERROR=TRUE
470 ....ATTRIBUTE(MAP,CUST_NUMBER)="CUR,BRI"
480 ....MESSAGE="CUSTOMER NUMBER MUST BE 6 CHARACTERS"
490 ...ELSE
```

```
500 ....GET RECX(CUST_NUMBER)
510 ....IF RECX="FOUND"
520 .....ERROR=TRUE
530 .....ATTRIBUTE(MAP,CUST_NUMBER)="CUR,BRI"
540 .....MESSAGE="DUPLICATE CUSTOMER NUMBER--PLEASE CHANGE"
550 ....END
560 ...END
570 ..END
580 .END
590 EXIT
```

Now, run the program and enter a customer number that already exists in the file:

```
                              B U R R Y S
                          NEW CUSTOMER ENTRY


                NAME:                       BOB GREEN
                ADDRESS:                    1 MAIN STREET
                CITY:                       CINCINNATI
                STATE:                      OH
                ZIP CODE:                   45000

                CUSTOMER NUMBER:            333333
                CLASS:                      10
                CUSTOMER CREDIT RATING:     AA
                CREDIT LIMIT:               5000
                BRANCH NUMBER:              1234
                COMMENTS:

DUPLICATE CUSTOMER NUMBER--PLEASE CHANGE
```

Next, change the CUST_NUMBER to one that doesn't already exist in the file and press ENTER. MANTIS inserts the record into CUST_INFO and clears the error message:

```
                    B U R R Y S
                NEW CUSTOMER ENTRY


NAME:
ADDRESS:
CITY:
STATE:
ZIP CODE:

CUSTOMER NUMBER:
CLASS:
CUSTOMER CREDIT RATING:
CREDIT LIMIT:
BRANCH NUMBER:
COMMENTS:
```

Press CANCEL to chain to the menu program. Enter 2 in the action field to display the customer browse screen. You'll find the customer you just entered in the file.

Step 12: Enhancing the customer entry program

When you write programs, flexibility and ease of maintenance are important considerations. For instance, suppose you wanted to change the attributes of fields on your screen when an error condition occurs. To do that, you would have to alter “CUR,BRI” in four places (lines 360, 420, 470, and 530). However, if you replace the text literal “CUR,BRI” with a variable, you can make subsequent changes to the field attributes by changing only a single line in your program.

Use the TEXT statement to name a variable and allocate memory locations for it. With TEXT, you must supply a symbolic name and, optionally, the maximum length (in parentheses) for your variable:

TEXT ***name1*** [([***n*,**] ***length***)]
[**,*name2*** ([***n*,**] ***length***) **. . .**]

Enter the following statement on the command line and press ENTER:

```
105 TEXT HIGHLIGHT_ERROR(20)
```

Now the variable HIGHLIGHT_ERROR is valid for this program, and can hold a text value up to 20 characters long.

Next, replace the text literal, “CUR,BRI” in your subroutine with the variable HIGHLIGHT_ERROR. Instead of scanning the program for ATTRIBUTE statements, you can issue the USAGE command

USAGE ATTRIBUTE

On the same line where you declared the text variable HIGHLIGHT_ERROR, you can set it equal to the attributes you specified earlier. Use a colon (:) to separate commands that appear on the same line. Change line 105 as shown:

```
105 TEXT HIGHLIGHT_ERROR(20):HIGHLIGHT_ERROR="CUR,BRI"
```

Whenever possible, design program modules so that they display on a single screen, so that they’re easier to read. In CUST_ENTRY, neither the main module nor the subroutine meets this criterion. However, there is a way to shorten both routines.

Notice that we initialized ERROR at line 180, and switched it to TRUE four times in the subroutine. Since each edit check also returns a message when it detects an error, we can use the variable MESSAGE to perform the same task as ERROR does now.

Begin by setting MESSAGE equal to null in the subroutine. Enter the following on the command line and press ENTER:

```
335 MESSAGE=""
```

Next, make the INSERT at line 200 dependent on the value of MESSAGE, rather than ERROR. Change line 200, as shown:

```
200 IF MESSAGE=""
```

Now, MANTIS will insert records into the file only if the MESSAGE field is blank; that is, if there are no errors on your screen. This means that you no longer need the variable, ERROR, or the switches in your program. Therefore, ERASE statements 180, 350, 410, 460, and 520. Renumber the program again and replace it.

Compare your version with this listing:

```
100 ENTRY CUST_ENTRY
110 .TEXT HIGHLIGHT_ERROR(20):HIGHLIGHT_ERROR="CUR,BRI"
120 .SCREEN MAP("CUST_ENTRY")
130 .FILE REC("CUST_INFO","PASSWORD")
140 .FILE REC2("STATE_CODES","PASSWORD")
150 .FILE RECX("CUST_INFO","PASSWORD",PREFIX)
160 .CONVERSE MAP
170 .WHILE MAP<>"CANCEL"
180 ..WHEN MAP<>"PF1"
190 ...DO VALIDATE_CUST_INFO
200 ...IF MESSAGE=""
210 ....INSERT REC
220 ....CLEAR MAP
230 ...END
240 ..WHEN MAP="PF1"
250 ...PROMPT"STATE_CODES"
260 ..END
270 ..CONVERSE MAP
280 ..ATTRIBUTE(MAP)="RES"
290 .END
300 .CHAIN"CUST_MENU"
310 EXIT
320 |
330 ENTRY VALIDATE_CUST_INFO
340 .MESSAGE=""
350 .IF SIZE(CUST_NAME)=0
360 ..ATTRIBUTE(MAP,CUST_NAME)=HIGHLIGHT_ERROR
```

```
370 ..MESSAGE="CUSTOMER NAME MISSING--PLEASE ENTER"
380 .ELSE
390 ..GET REC2(CUST_STATE)EQUAL
400 ..IF REC2<>"FOUND"
410 ...ATTRIBUTE(MAP,CUST_STATE)=HIGHLIGHT_ERROR
420 ...MESSAGE="INCORRECT OR MISSING CODE--PRESS 'PF1' FOR HELP"
430 ..ELSE
440 ...IF SIZE(CUST_NUMBER)<6
450 ....ATTRIBUTE(MAP,CUST_NUMBER)=HIGHLIGHT_ERROR
460 ....MESSAGE="CUSTOMER NUMBER MUST BE 6 CHARACTERS"
470 ...ELSE
480 ....GET RECX(CUST_NUMBER)
490 ....IF RECX="FOUND"
500 ....ATTRIBUTE(MAP,CUST_NUMBER)=HIGHLIGHT_ERROR
510 ....MESSAGE="DUPLICATE CUSTOMER NUMBER—PLEASE CHANGE"
520 ....END
530 ...END
540 ..END
550 .END
560 EXIT
```

Because you have removed four statements from CUST_ENTRY, you can now view statements 330-530 (including all four edit checks) on a single screen.

Exercise

Design a maintenance program that will read particular records from the Burrys customer file and delete or update them. Name the program CUST_MAINT.

Write your DELETE and UPDATE statements in the following formats:

```
DELETE file-name [LEVEL=n]
UPDATE file-name [LEVEL=n]
```

Your program should display the Burrys new customer screen and accept a customer number from the user. If the number exists in the file, MANTIS should display the record and the following message:

```
'PF1' TO DELETE, 'PF2' TO UPDATE, 'PF3' TO CANCEL
```

The program should then perform the following actions:

- If a user presses PF1, MANTIS should delete the record, clear the screen, and return “DELETION COMPLETE.”
- If a user presses PF2, MANTIS should update the record, clear the screen, and return “UPDATE COMPLETE.”
- If a user presses PF3, the program should clear the screen and return “MAINTENANCE CANCELLED AT USER’S REQUEST--CANCEL TO EXIT.”
- If MANTIS cannot find a user’s entry, it should clear the screen and return “CUSTOMER NOT FOUND”

All of the commands you need to write CUST_MAINT appear in CUST_BROWSE. Our version of the CUST_MAINT program appears in “Exercise examples” on page 189.

9

What's next?

You've now completed a basic course in MANTIS application development. What's your next step?

First, don't hesitate to go back and redo any of the lessons or exercises, especially those that seemed difficult to you. You'll be surprised how much easier they are the second time through.

Second, you can find more information on MANTIS programming in AD/Advantage MANTIS Language OpenVMS/UNIX, P39-1310.

For more information on the MANTIS design facilities (screen design, file design, etc.), refer to AD/Advantage MANTIS Facilities OpenVMS/UNIX, P39-1300.

Finally, a Reader Comment Sheet appears at the end of this tutorial. If you have any comments on the tutorial, or suggestions for more advanced applications, please take a moment to complete this form and send it to us.

A

Exercise examples

This appendix provides our solutions for the exercises in the preceding chapters.

Chapter 1: Introduction

No exercises were presented in chapter 1.

Chapter 2: Creating a screen

In chapter 2, you performed four exercises:

- Creating the customer accounts menu
- Creating the new customer entry screen
- Creating the state code entry screen
- Viewing the directory of screens

Exercise 1: Creating the customer accounts menu

The customer accounts menu should look like this:

```
B|U|R|R|Y|S
CUSTOMER|ACCOUNTS|SYSTEM


ENTER|A|NEW|CUSTOMER||.............|1

VIEW|CUSTOMER|BROWSES|SCREEN||.....|2

EXIT|THIS|FACILITY||...............|CANCEL

: # :
```

It contains the following field:

Heading	Symbolic name	Field length	Attributes
None	ACTION	1	Numeric

Exercise 2: Creating the new customer entry screen

The new customer entry screen should look like this:

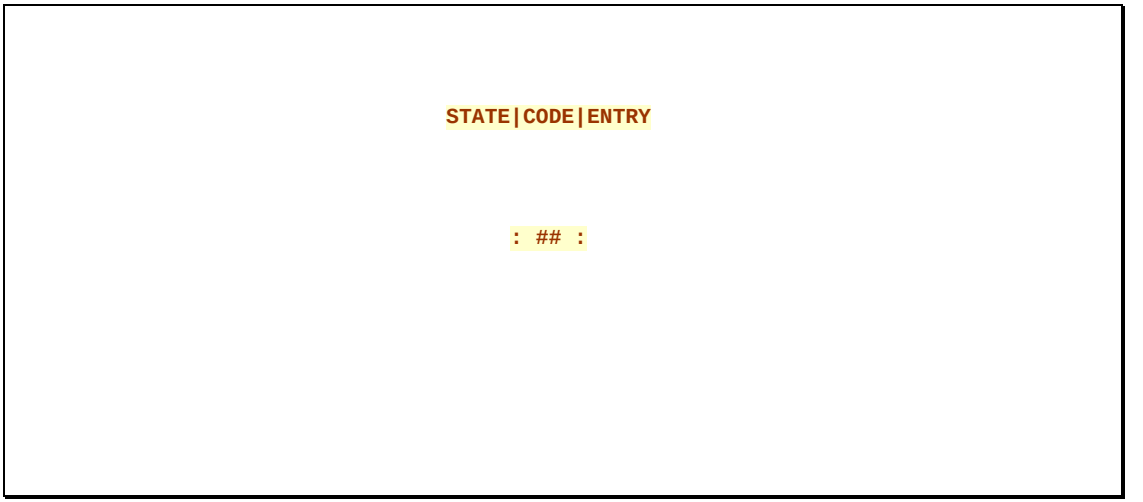
```
                    B|U|R|R|Y|S
                NEW|CUSTOMER|ENTRY


          NAME:                     ####################
          ADDRESS:                  ####################
          CITY:                     #############
          STATE:                    ##
          ZIP|CODE:                 #####

          CUSTOMER|NUMBER:          ######
          CLASS:                    ##
          CUSTOMER|CREDIT|RATING:   #####
          CREDIT|LIMIT:             #####
          BRANCH|NUMBER:            ######
          COMMENTS:                 ########################

################################################################################
```

It contains the following fields:

Heading	Symbolic name	Field length	Attributes
NAME	CUST_NAME	20	Autoskip
ADDRESS	CUST_ADDRESS	20	Autoskip
CITY	CUST_CITY	13	Autoskip
STATE	CUST_STATE	2	Autoskip
ZIP CODE	CUST_ZIP_CODE	5	Numeric, autoskip
CUSTOMER NUMBER	CUST_NUMBER	6	Autoskip
CLASS	CUST_CLASS	2	Autoskip
CUSTOMER CREDIT RATING	CUST_CREDIT_RAT	2	Autoskip
CREDIT LIMIT	CUST_CREDIT_LIM	5	Numeric, autoskip
BRANCH NUMBER	CUST_BRCH_NUMBER	4	Autoskip
COMMENTS	CUST_COMMENTS	25	Autoskip
	MESSAGE	79	Bright, protected

Exercise 3. Creating the state code entry screen

The state code entry screen should look like this:

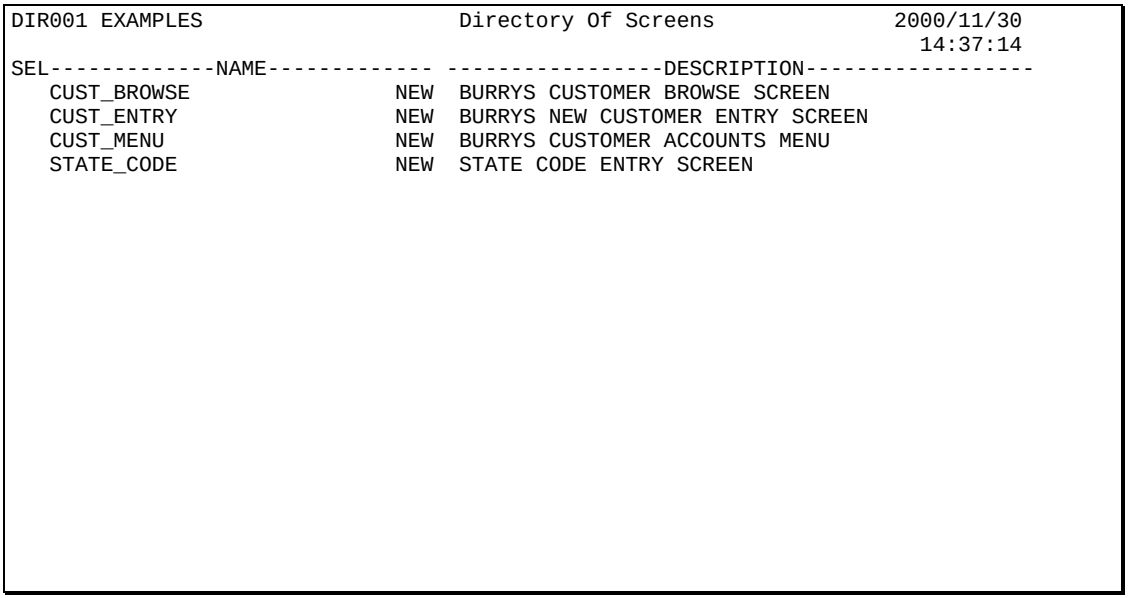


It contains the following field:

Heading	Symbolic name	Field length	Attributes
None	CUST_STATE	2	None

Exercise 4: Viewing the directory of screens

When you have finished designing and saving the four screens for the Burrys scenario, your Directory of Screens should contain the following screens:



```
DIR001 EXAMPLES                    Directory Of Screens             2000/11/30
                                                                      14:37:14
SEL-------------NAME------------ -----------------DESCRIPTION-----------------
   CUST_BROWSE                NEW  BURRYS CUSTOMER BROWSE SCREEN
   CUST_ENTRY                 NEW  BURRYS NEW CUSTOMER ENTRY SCREEN
   CUST_MENU                  NEW  BURRYS CUSTOMER ACCOUNTS MENU
   STATE_CODE                 NEW  STATE CODE ENTRY SCREEN
```

Chapter 3: Creating a MANTIS file

The record layout for the STATE_CODES file should look like this:

```
MFV003                 MANTIS Record Layout Definition              YYYY/MM/DD
Name: STATE_CODES                                                   HH:MM:SS
     Page 1                     Element Count 1                 Size 32
     Element  ------Name------  Data-type  Dimensions  ----Attributes----

         1    CUST_STATE        TEXT       2           KEY




              (Use PF1 - PF4 to page; use CANCEL to exit)
```

Chapter 4: Creating a prompter

No exercises were presented in chapter 4.

Chapter 5: Understanding MANTIS programming fundamentals

In the exercises for this chapter, you:

- Verified the menu program to ensure that you entered it correctly
- Created a program to display a screen and add records to the state codes file
- Created a program to display a screen and add records to the customer information file

Exercise 1: Verifying the menu program

The menu program you created in chapter 5 should look like this:

```
100 ENTRY CUST_MENU
110 .SCREEN MAP("CUST_MENU")
120 .CONVERSE MAP
130 .WHILE MAP<>"CANCEL"
140 ..WHEN ACTION=1 OR MAP="PF1"
150 ...SHOW"PROGRAM CHAINS TO CUSTOMER ENTRY SCREEN"
160 ...WAIT
170 ..WHEN ACTION=2 OR MAP="PF2"
180 ...SHOW"PROGRAM CHAINS TO CUSTOMER BROWSE SCREEN":WAIT
190 ..END
200 ..CLEAR MAP
210 ..CONVERSE MAP
220 .END
230 EXIT
```

Exercise 2: Adding records to the state codes file

Your program to add records to the state codes file should look like this:

```
100 ENTRY STATE_CODES
110 .SCREEN MAP("CUST_STATE")
120 .FILE REC("STATE_CODES","PASSWORD")
130 .CONVERSE MAP
140 .WHILE MAP<>"CANCEL"
150 ..INSERT REC
160 ..CLEAR MAP
170 ..CONVERSE MAP
180 .END
190 EXIT
```

Exercise 3: Adding records to the customer information file

Your program to add records to the customer information file should look like this:

```
100 ENTRY ADD_CUST
110 .SCREEN MAP("CUST_ENTRY")
120 .FILE REC("CUST_INFO","PASSWORD")
130 .CONVERSE MAP
140 .WHILE MAP<>"CANCEL"
150 ..INSERT REC
160 ..CLEAR MAP
170 ..CONVERSE MAP
180 .END
190 EXIT
```

Chapter 6: Using MANTIS programming statements and commands

In chapter 2, you created the stubs for two programs, CUST_ENTRY and CUST_BROWSE.

Exercise 1: Creating a stub for the customer entry program

Your stub for the CUST_ENTRY program should look like this:

```
100 ENTRY CUST_ENTRY
110 .SCREEN MAP("CUST_ENTRY")
120 .CONVERSE MAP
130 .WHILE MAP<>"CANCEL"
140 ..CONVERSE MAP
150 .END
160 .CHAIN"CUST_MENU"
170 EXIT
```

Exercise 2: Creating a stub for the browse program

Your stub for the CUST_BROWSE program should look like this:

```
100 ENTRY CUST_BROWSE
110 .SCREEN MAP("CUST_BROWSE")
120 .CONVERSE MAP
130 .WHILE MAP<>"CANCEL"
140 ..CONVERSE MAP
150 .END
160 .CHAIN"CUST_MENU"
170 EXIT
```

Chapter 7: Creating a browse program

In chapter 7, you created the CODE_BROWSE program, then enhanced it.

Exercise 1: Writing a program to browse the state codes file

Your first version of the CODE_BROWSE program should look like this:

```
100 ENTRY CODE_BROWSE
110 .SCREEN MAP("STATE_CODE")
120 .FILE REC("STATE_CODES","PASSWORD")
130 .WHILE REC<>"END"AND MAP<>"CANCEL"
140 ..GET REC
150 ..CONVERSE MAP
160 .END
170 EXIT
```

Exercise 2: Enhancing the state codes browse program

Your enhanced version of the CODE_BROWSE program should look like this:

```
100 ENTRY CODE_BROWSE
110 .SCREEN MAP("STATE_CODE")
120 .FILE REC("STATE_CODES","PASSWORD",14)
130 .WHILE REC<>"END"AND MAP<>"CANCEL"
140 ..CLEAR MAP
150 ..BUFFER=BUFFER+1
160 ..GET REC LEVEL=BUFFER
170 ..WHILE REC<>"END"AND BUFFER<14
180 ...BUFFER=BUFFER+1
190 ...GET REC LEVEL=BUFFER
200 ..END
210 ..CONVERSE MAP
220 .END
230 .CHAIN"CUST_MENU"
240 EXIT
```

Chapter 8: Creating a data entry program

In chapter 8, you designed a new maintenance program, CUST_MAINT. CUST_MAINT uses many of the same commands and structures as CUST_BROWSE. The TEXT statement establishes the variable, MSG, and allows it to contain a message up to 60 characters long. The ATTRIBUTE statement locates the cursor at the beginning of the CUST_NUMBER field.

Within the WHILE-END loop (160-360), MANTIS first tries to retrieve an exact key match for the customer number that a user enters. There are only two possible answers: either the record exists in the file, or it doesn't. Therefore, an IF-ELSE-END structure follows. The IF block consists of WHEN structures that allow the user to choose any one of three options: delete, update, or cancel.

Statements 330-350 clear a user's entry from the screen, set the MESSAGE field equal to the text literal contained in MSG, and prepare the screen for a user's next entry.

The completed CUST_MAINT program should look like this:

```
100 ENTRY CUST_MAINT
110 .SCREEN MAP("CUST_MENU")
120 .FILE REC("CUST_INFO","PASSWORD")
130 .TEXT MSG(60)
140 .ATTRIBUTE(MAP,CUST_NUMBER)="CUR"
150 .CONVERSE MAP
160 .WHILE MAP<>"CANCEL"
170 ..GET REC(CUST_NUMBER)EQUAL
180 ..IF REC="FOUND"
190 ...MESSAGE="'PF1' TO DELETE, 'PF2' TO UPDATE, 'PF3' TO CANCEL"
200 ...CONVERSE MAP
210 ...WHEN MAP="PF1"
220 ....DELETE REC
230 ....MSG="DELETION COMPLETE"
240 ...WHEN MAP="PF2"
250 ....UPDATE REC
260 ....MSG="UDATE COMPLETE"
270 ...WHEN MAP="PF3"
280 ....MSG="MAINTENANCE CANCELLED AT USER'S REQUEST--CANCEL TO EXIT"
290 ...END
300 ..ELSE
310 ...MSG="CUSTOMER NOT FOUND"
320 ..END
330 ..CLEAR MAP
340 ..MESSAGE=MSG
350 ..CONVERSE MAP
360 .END
370 .CHAIN"CUST_MENU"
380 EXIT
```

Chapter 9: What's next?

No exercises were presented for chapter 9.

Index

E

F

G

H

I

K

L

M

N

O

P

R

S

T

U

V

W